昭和61年1月1日発行（毎月1回1日発行）第5巻1号通巻45号　昭和60年5月14日国鉄首都特別扱承認雑誌8269号　昭和58年11月2日第三種郵便物認可

ISSN 0910-7614

MZ,X1 & ポケコンシリーズ

Oh!MZ

パソコン情報誌
PERSONAL COMPUTER MAGAZINE

特集 BASIC"行動学"入門
付・N88/F/MSX BASIC対照表

THE SOFTOUCH SPECIAL
GAME OF THE YEAR
ノミネート発表

新年特別増大号

S-OS全機種共通システム
FM音源サウンドエディタ

新連載 ビジネスソフトへの対応
Multiplanの基礎知識

新連載 turbo LOGO入門
タートルグラフィックの世界

MZ-1500回路図エディタ
THE 部品箱

X1/X1turbo CG Gallery
カラーイメージボード

1
JAN.1986
定価480円

圧倒的なスピードと色彩感覚、高精細度グラフィックス

ゲームに実務にすばらしい**スピード**を実現。
また、**640×400ドット**(標準で4色、最大16色)、
256色同時表示(320×200ドットモード)のきわだつカラー表現、
別売のカラーパレットボードを使えば4096色のうち
任意の15色を表示できます。

ナレーションを入れたり、楽器音もつくれます。

自分の声や電話からの声を吹き込んだり、音楽も聞ける**ボイスレコーダ**を搭載。
※**ボイスメール**や※留守番電話といったテレコミュニケーションに利用できるほか、
ナレーションを入れた個性あるソフトが自由につくれます。
また、きれいな**FM音源**(8オクターブ3重和音)も採用しました。
※テレホンソフトの通信機能を活用するためには別売のモデムホンが必要です。

日本語がパワーアップ、JIS第2水準漢字ROMも標準装備。

JIS第1水準に加えて**第2水準漢字**もサポート。
漢字BASIC(M25/S25)を採用したきわだつ日本語処理機能。
プログラム中の変数や配列名、ラベル名などに漢字が使え、変換もスピーディ。
別売の辞書ROMボードを使えば**文節変換**もOKです。

●コンパクトながら大容量640KBの3.5インチマイクロフロッピー搭載(Model 20/1基・Model 30/2基)●最大256KBの大容量メモリ(標準128KB)●カナの50音配列も可能な多機能キーボード●BASICやテレホンソフト使用時にも電卓機能などの特殊機能を利用できる便利なアルゴ(割り込み)機能●スイッチひとつで切替えできるMZ-2200/2000モード、MZ-80Bモードを装備して多くの資産を継承く主な別売品>■14型カラーディスプレイMZ-1D22標準価格108,000円■モデムホンMZ-1X19標準価格98,000円■ボイスコミュニケーションインターフェイスMZ-1E26標準価格24,800円■80桁漢字ドットプリンタMZ-1P18標準価格188,000円■マウスMZ-1X10標準価格19,800円■カラーパレットボードMZ-1M10標準価格14,500円■辞書ROMボードMZ-1R28標準価格22,000円■PERSONAL CP/M™※1(WORDMASTER™※2付)MZ-6Z001標準価格16,800円

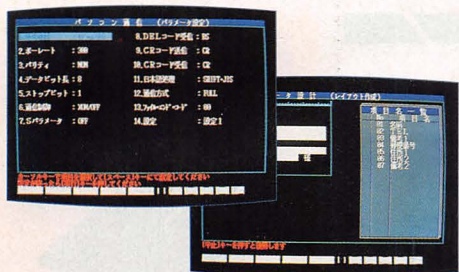
8ビットパーソナルコンピュータMZ-2500シリーズ

Model 20(MZ-2511・640KB3.5″FD1基付)標準価格168,000円
Model 30(MZ-2521・640KB3.5″FD2基付)標準価格198,000円

シャープ株式会社 本社 〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表) ●お問い合わせは…本社内国内情報システム営業本部まで

◀「テレホンソフト」のメインメニュー。時刻表示とともに6つの処理内容が表示され、ディファイナブルファンクションキーにより仕事を選択します。

▶データベースで作成した電話帳の表示・検索を指定した画面。TABキーによって検索項目が変更できます。

▼パソコン通信の通信パラメータ設定。設定は全部で13項目。最終設定が自動的にディスクに記録されます。

▲作成したデータをプリンタに印刷するためのレイアウトも作成できます。

視線は、いま「スーパーMZ」。

本格データベース機能がうれしい〝テレホンソフト〟標準装備。

話題のBBS(電子掲示板)にアクセスできる**ターミナル機能**や**データ通信機能**に加え、**登録件数最大4,000件**の本格的な**カード型データベース機能**を装備。住所録から顧客の管理まで幅広く使えてとても便利。ハードの凄さに応えたすばらしいソフトです。

※テレホンソフトの通信機能を活用するためには別売のモデムホンが必要です。

写真はModel30です。写真の14型カラーディスプレイ、モデムホンは別売です。また本体に装着されているカセットテープは撮影用で、付属品・市販品ではありません。画面はハメコミ合成です。

MZ-2500
資料請求券
Oh!Mz・1月

Oh!MZ JANUARY 1986 1

CONTENTS

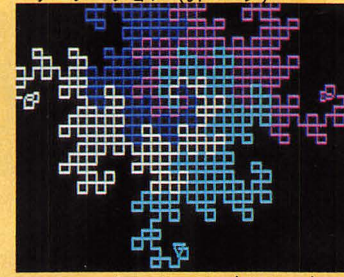
表紙絵：MORIHO TANAKA

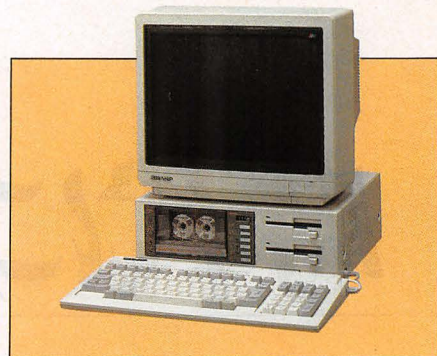
特集

カラー

THE SOFTOUCH

▲カラーイメージボード（30ページ）

▲SUPER RAM BORDによるアニメーション（31ページ）

▲タートルグラフィックス（33ページ）

シャープ パーソナルコンピュータ MZ-2500
コンピュータ本体 MZ-2521 198,000円
MZ-2511 168,000円
カラーディスプレイMZ-1D22 108,000円

THE部品箱（182ページ）▲

ドルアーガの塔（138ページ）▲

リザード（136ページ）▲

■広告目次

<スタッフ>
●編集長——安田千尋 ●編集——前田 徹 土平章博 永野 仁 菊川良子 三上之彦 ●協力——有田隆也 高野庸一 西畑文広 Itti Rittaporn 河本恭彦 清水和人 後藤貴行 林 一樹 斎藤 亮 近藤弘幸 浅野恵造 工藤 誠 茗原秀幸 小森 隆 挙市哲司 ●カメラ——浜崎 昭 杉山和美 ●イラスト——永沢しげる 山田晴久 ●アートディレクター——中島真子 ●レイアウト——CAN ART 元木昌子 中島由紀子 ●校正——グループごじら

疾走するグラフィックス

XⅠF発進。

Model 20（ミニフロッピーディスクドライブ 1ドライブ内蔵）

Model 10（高速電磁メカ カセットデータレコーダ内蔵）

パソコンテレビ

- ●Model 10 パーソナルコンピュータ＋キーボード CZ-811CE（オフィスグレー）・R（ローズレッド） 標準価格 89,800円
- ●Model 20 パーソナルコンピュータ＋キーボード CZ-812CE（オフィスグレー）・R（ローズレッド） 標準価格139,800円
- ●14型カラーディスプレイレイテレビ CZ-811DE（オフィスグレー）・R（ローズレッド） 標準価格 89,800円

SHARP

for X1

X1turbo用 システム・ユーザー辞書

日本語処理機能、いよいよ充実。

X1ターボの標準BASICとの併用により熟語変換が可能な「システム辞書」と、ユーザーが自由に文字を登録でき自分専用のオリジナル辞書がつくれる「ユーザー辞書」のユーティリティからなるソフトウェアです。システム辞書には、標準BASICの音訓辞書のほとんどすべてと、日常よく使われる熟語、人名、地名など約3万語が収録されており、さらに辞書変更用ユーティリティを利用すれば、内容の追加、書き換えもOK。またユーザー辞書は、自分専用の辞書としての活用の他、住所録や電話帳などにも応用できます。

■2D・5″FD版CZ-111SF　標準価格8,800円

X1turbo用 グラフィックツール 嬉楽画（きらくが）ターボ

スーパーインポーズ画像作成に、ビデオ編集に。

わかりやすいアイコン表示で、プログラムの組めない初心者の方にも、複雑なコンピュータ画像を楽しみながら手軽に作画できるうれしいグラフィックツールです。入力は、マウスでとっても簡単。精密400ラインモードも装備しています。さらにビデオマルチプロセッサ（CZ-8VP1）の入力切換えをコントロールできるタイムテーブル機能を装備。ビデオ編集にたいへん便利です。

〈アイコン表示によるグラフィックコマンド〉
●ライン●ボックス●ボックスフル●サークル●ペイント●スプレー●ブラシ●パレット●ルーペ

■2D・5″FD版（Model 20、30、40用）
CZ-114SF（マウス付）標準価格17,800円

X1turbo用 turboLOGO（漢字版）

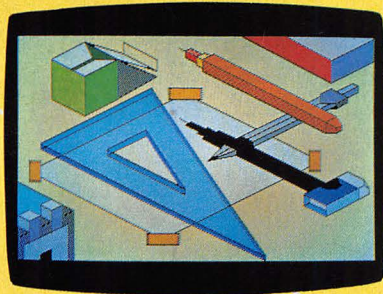
ヒューマンなLOGOでターボは進化する。

新時代の言語LOGOがいよいよターボで走ります。絵やグラフ、模様などを簡単な命令でわかりやすくプログラミングできるタートルグラフィックス機能をはじめ、構造化プログラミング機能、優れたリスト処理機能など、BASICなどの言語にはない独自の機能を持つLOGO――とりわけこのturbo LOGO（漢字版）は、プロシジャーや変数、データに漢字をサポート。日本語LOGOとしてのやさしさに加え、マウスを使って絵を書いたり、プログラミングもこなせる多機能ぶり。このヒューマンなソフトウェアによってあなたの知的創造の世界はさらに拡がります。

■2D・5″FD版CZ-117SF 標準価格18,800円

いずれがキャビアかフォアグラか。

いわばアピタイザーからデザートまで、メーンディッシュのX1をひきたてる
ピリッと効いたおいしいソフトたち。
いま洗練のハードに応えて、オリジナルソフトの輪がどんどん拡がっています。

X1シリーズ用 NEW BASIC(Version2.0)

ターボに迫る高速グラフィックを実現。

X1シリーズに待望のニューBASICが登場しました。まさにターボなみの高速グラフィック。高速ペイントルーチンの採用で、ペイント速度は従来の約35倍（X1 BASIC V1.0との比較）。さらにX1ターボで好評のNEW ON命令によるBASICの10段階カット機能も導入、ユーザーエリアを拡大できます。また漢字機能もパワーアップ。漢字ユーティリティのサポートで漢字の扱いがさらに身近に。まさにX1ユーザーには見逃がせないBASICです。

■カセット版　CZ-112SF 標準価格7,800円
■2D・3″FD版 CZ-113SF 標準価格8,800円
■2D・5″FD版 CZ-124SF 標準価格8,800円

X1 turbo用 Multiplan™

表計算用簡易言語として高い評価を得ているあの「Multiplan」がターボで走ります。計算・作表のための豊富な機能に加えて、操作しやすいコマンドメニュー方式、高度な日本語処理など、高機能と使いやすさを実現したビジネスソフトです。また増設RAM(64KB)の使用により、処理スピードを早めています。

■2D・5″FD版（Model 20、30、40用）
CZ-127MF 標準価格49,800円

●このソフトの使用にあたっては2D・5″FDが2基必要です。
※Multiplanは米国マイクロソフト社の登録商標です。

X1 turbo用 turboターミナル

「TeleStar」や「アスキーネットワーク」など、話題のネットワークにアクセスしたり、パソコン間のデータ通信（漢字対応）がスピーディに楽しめる通信ソフト。モデム付電話を使用した場合には自動発信/自動着信も可能です。

■X1 turboシリーズ用 5″・2D FD版
CZ-131SF 標準価格 8,800円

● RS-232C用ケーブル CZ-8LM1（平行接続型）/CZ-8LM2（クロス接続型）各標準価格7,200円
※公衆回線を使って通信する場合、モデム付電話か音響カプラが必要です。

X1・X1 turbo用 ランゲージシリーズ

■各2D・5″FD版 各標準価格13,800円

科学技術計算の分野に適した高級言語
FORTRAN (CZ-115LF)

事務分野で威力を発揮する伝統の言語
COBOL (CZ-118LF)

人工知能研究の中心的言語
LISP (CZ-120LF)

いま熱い視線を集めるC言語
C (CZ-116LF)

話題の人工知能言語
PROLOG (CZ-119LF)

拡張性に優れたスクリーンエディット型言語
FORTH (CZ-121LF)

系統的プログラミング設計に適した言語
PASCAL (CZ-125LF)

文法が明快な数学的プログラミング言語
APL (CZ-126LF)

ランゲージマスター(CP/M®)

■2D・5″FD版 CZ-128SF 標準価格 9,800円

※CP/Mは米国デジタルリサーチ社の登録商標です。
▶ランゲージシリーズの使用にあたっては、ランゲージマスターまたはCZ-5CPMが必要です。

人気のゲーム、ワープロソフトなど
家族そろって楽しめるソフトの6本セット

X1シリーズソフトウェアパック The YOKOZUNA

●テープ版　CZ-122PF　標準価格 15,800円
●5インチディスク版 CZ-123PF　標準価格 19,800円

シャープ株式会社●お問い合わせは…シャープ㈱電子機器事業本部ソフト開発部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。

資料請求券
X1ソフト
oh!MZ
12係

ワープロのスタンダード

ライバルは16ビット

日本語ワードプロセッサー

〈即戦力〉

SHARP X1・X1turbo用
新発売　¥55,000

NEC PC-8801mkII SR/TR/FR/mR用
新発売　¥55,000

これが、8ビットワープロの新しい規準

8ビットマシーンの能力を限界まで引きだした、卓越のテクノロジー

〈即戦力〉が、8ビットマシーンの潜在能力を、見事に目覚めさせます。ハイレベルな変換効率を誇り 登録済4万語、熟語・短文・外字登録の充実の辞書機能、最高速の変換スピード、倍角や¼角等の豊富な表現力、移動や複写さらに検索や置換え等の強力な編集機能、多彩な印刷機能等、これまでの8ビットワープロでは考えられなかったパフォーマンスを実現しています。しかも、初めての方でもディスプレイ画面に表示されるガイドにより簡単に操作できますから、まさに導入と同時に即、戦力として活用できます。優しく入門できて、使いこむほど高機能を発揮する〈即戦力〉、これこそ8ビットワープロの新しい規準です。

キーに慣れるのが、あなたの仕事あとは〈即戦力〉がフォローします。

ほんとうに使いやすいワープロは、どうあるべきか。サムシンググッドからの、回答です。

単にできる機能があるということと、本当に使えるということは、違うとサムシンググッドは考えます。私達は、一つ一つの機能をほんとうに使えるところまで質を高めたうえで、はじめて搭載しています。例えば、辞書機能一つとっても、登録語数の多さだけでなく、その内容を充実させています。ビジネス文書や、新聞、雑誌、小説、論文等から「活きている言葉」を集録しています。この質に加え、4万語という膨大な量を持つことで、ハイレベルな変換効率を可能にしたわけです。ここに〈即戦力〉の使いやすさの、最大の理由があります。私達は「機能の質」ともいうべき高機能と、「機能の量」である多機能の同時追求が、使いやすいワープロの条件であると信じます。高機能を積上げて、多機能を創りあげる、これがサムシンググッドの基本です

主な仕様

●**付属品**／15分間マニュアル●**文例集**／ビジネス文書ディスク（ビジネス文書50例登録）●**漢字**／JIS第一水準・JIS第二水準文字●**文字種**／全角、半角（英数字カタカナ）、倍角、¼角（英数字）、上つき、下つき文字●**外字**／40字（16×16ドットまたは24×24ドット）●**画面制御**／上下スクロール、左右スクロール（最大82字）、前画面・次画面表示、頁指定・文頭・文末呼び出し、頁・行・桁位置表示●**印刷**／印刷枚数・用紙サイズ・印刷範囲・横書・縦書・一頁行数・一行文字数、文字間隔・改行幅（用紙の大きさにより自動設定）、差し込み印刷、宛名書き印刷、袋とじ印刷●**文書管理**／文書名登録（かな漢字まじり15文字まで）、文書名変更、文書名一覧表表示、文書複写、文書削除、文書併合●**辞書**／5"2D登録済4万語以上+ユーザー登録8,000語（40K）●**レイアウト**／中央寄せ・右寄せ・左寄せ、下線、網かけ、改行・改頁●**プリンター機種**／SHARP／CZ-800P、CZ-8PD2、CZ-80PK、CZ-8PK2、CZ-8PN1、MZ-1P03、MZ-1P06、MZ-1P07、MZ-1P08、MZ-1P10、MZ-1P11、MZ-1P14、MZ-1P10A、MZ-1P11A／EPSON／RP-80、RP-80K、RP-80II、RP-80IIK、RP-80F／TII、RP-80F／TIIK、FP-80、FP-80K、UP-130K（ESC／P、PC）、IP-130K（ESC／P、PC）／NEC／PC-PR101、PC-PR201、PC-8822
●**ユーティリティー**／文書ディスクのフォーマット・バックアップ・コピー、辞書ディスクのバックアップ、文書一覧表の印刷●**短文登録**／16個（読み方最大12文字、短文最大120文字）
※本仕様はSHARP X1・X1turbo用のものです

16ビットシリーズ好評発売中！
NEC PC-9801用
〈即戦力〉・〈即戦力スーパー〉
IBM JX3/4用〈即戦力〉

人を大切にするテクノロジー
株式会社 サムシンググッド
〒160 東京都新宿区大久保2-5-20シティプラザ新宿3F TEL03(232)0801

世界初の24ドットインテリジェント

①市販の日本語ワ

②はがき印字が

③定型書式印字

① PC-88、98シリーズに対応する日本語ワープロソフト(ユーカラ、テラ…)顧客管理ソフト(駿漢、新漢客…)など、あらゆる市販ソフトが使えます。

●NEC NM-9300Sとコンパチブル。PC-PR201にも対応。
●MSX対応24ドット漢字プリンター。

PC-88-98シリーズ対応

MSX機に対応

② はがきの印字フォーマットをプリンターが内蔵しているので、市販の日本語ワープロソフト・顧客管理ソフトなどにより定位置にカンタンに印字できます。

Ⓐ 〒104-01
東京都中央区京橋3-3-8
新京橋ビル3F
ブラザー販売株式会社
情報機器事業部　営業部
PUB　事務局

Ⓑ PUBは、ブラザープリンターをご使用頂いているユーザーの友の会です。
会員の方には、PUB MEDIAが定期に送られます。

〒438-02
静岡県磐田郡豊岡村松島103番地
松島アパート3棟　308号
村岡　幾三　様

(備　考　欄)
本　　体：PC-8801SR
プリンター：HR-6X
生年月日：昭和29年11月12日

〒108-03
東京都港区三田3-20-18
三田駅前ビル　25F
東京ソフト開発株式会社
開発部　設計5課
三田　太郎　様

(備　考　欄)
本　　体：PC-9801F
プリンター：M-1024P
生年月日：昭和26年10月8日

差出人住所データ

宛先人住所データ No.1

宛先人住所データ No.2

郵便はがき
108-03
東京都港区三田3-20-18
三田駅前ビル　25F
東京ソフト開発株式会社
開発部　設計5課
Ⓒ 三田　太郎　様
〒104-01
東京都中央区京橋3-3-8
新京橋ビル3F
ブラザー販売株式会社
情報機器事業部　営業部
PUB　事務局

郵便はがき
438-02
静岡県磐田郡豊岡村松島103番地
松島アパート3棟　308号
Ⓒ 村岡　幾三　様
〒104-01
東京都中央区京橋3-3-8
新京橋ビル3F
ブラザー販売株式会社
情報機器事業部　営業部
PUB　事務局

※用紙は官製はがき、もしくはそれに準ずる厚さのものをお使いください。

●まず郵便番号を。次に住所・氏名を頭ぞろえで連続インプットⒶ
●差出人、宛先人データは、漢字16文字×6行の範囲で自由にレイアウト。
●宛先人氏名は、見やすい縦倍角表記。Ⓒ
●ディップスイッチで縦でも横でも自由自在に印字可能、また差出人住所・氏名を印字しないこともできます。
●住所データの右側を備考欄として活用することもできます。Ⓑ

世界最小80桁シリアル9ドットインパクトプリンター

●各種アプリケーションソフト、又はハードコピーにてビットイメージで漢字が打てます。●ファンフォールド紙、ロール紙が使える3ウェイ・ペーパーハンドリング機能。●コピー枚数：オリジナル＋2P●印字速度：50文字/秒(M-1009)、40文字/秒(M-1009X)●重量：約3.0kg

写真は、M-1009X

FMシリーズ対応……………………MSX・PCシリーズ対応　¥49,800

M-1009&1009X

18ドット対応、熱転写漢字プリンター

●漢字が鮮やか、16×15ドット構成。●ほぼA4サイズのコンパクトボディ。●乾電池駆動で、機動性抜群。

オプション：
漢字ROMカートリッジ
(JIS第1水準)
HR-6X　¥30,000

MSX・PCシリーズ対応　¥49,800

brother

漢字プリンターだから
一プロソフトで、カンタン！もラクラク！

③ 官公庁提出書類、見積書、注文書など、すでに書式が印刷されている定型文書にもキメ細かくカンタンに印字できます。(キーボード使用の場合)

Ⓔ
Ⓓ NO.　NY60-123
会社名　日本商事株式会社
昭和　60年　10月　21日
件名　プリンターの見積りについて
納入先　貴社指定場所
納入場所　同　上
納入期日　受注後3日以内
荷造運賃　弊社負担
取引条件　従来通り
有効期限　昭和60年12月31日
担当　営業部　山田
総金額　¥13,940,000.－
NO. 1品名　HR-5/HR-5X
　価格　100　39,800　3980000
NO. 2品名　M-1009/M-1009X
　価格　100　49,800　4980000
NO. 3品名　HR-6X
　価格　100　49,800　4980000
NO. 4品名

(写真1)

御見積書　No. NY60-123
日本商事株式会社　御中　昭和60年10月21日
件名 プリンターの見積りについて
上記の件下記の通り御見積申し上げます。
何卒御用命の程御願い申し上げます。
名古屋市中区大須3丁目46番15号
ブラザー販売株式会社
情報機器事業部
納入先　貴社指定場所
納入場所　同　上
納入期日　受注後3日以内
荷造運賃　弊社負担
取引条件　従来通り
見積有効期間　昭和60年12月31日
担当　営業部　山田
総金額　¥13,940,000.－

品名及仕様	数量	単価	金額	記事
HR-5/HR-5X	100	39,800	3980000	
M-1009/M-1009X	100	49,800	4980000	
HR-6X	100	49,800	4980000	
合計				

- ●まず差込み印字データを頭ぞろえでインプット。Ⓓ
- ●キーボード(オプション)で、定型書式に沿って打たい位置を設定、登録します。(写真1)
- ●キーボードの記憶容量は487ヵ所。99分割が可能で、1ファイル最大60ヵ所。(バックアップ機能付)
- ●同時に3枚複写までできます。(ケミカルカーボン紙)用紙はA4。
- ●アンダーラインを引いた部分は、Ⓔ印字のとき無視されるため見出しとして使えます。(対応できないソフトもあります。)
- ●印字時間の短縮化を実現。

世界初！ブラザー発！

インテリジェント漢字プリンター

割付名人M-1024

NEW

- ●24ドットインパクト漢字プリンター
- ●高速漢字処理(20CPS→40CPS)
- ●気くばりの低騒音設計(減音モード付)
- ●しかも、小型・軽量・低価格
- ●もちろん、はがき・定型書式印字以外でも高性能発揮

この高機能で、この低価格！

M-1024P(PC-88、98対応 XIturbo、MZ-2500対応)……¥128,000
M-1024X(MSX対応)………¥128,000
フォーマットキーボードFK-20…………¥29,800
ピンフィードユニットPF-50……………¥5,000
JIS第2水準漢字ROMボード……………¥20,000
オートカットシートフィーダSF-20……1月発売予定

静かな印字、熱転写プリンター

●9ドット熱転写ビットイメージプリンター。●サイズ・重量：303(W)×65(H)×174(D)mm約1.6kg

写真は、HR-5X

FMシリーズ完全対応……MSX・PCシリーズ完全対応

¥39,800

PUB《Printer Users of Brother》会員募集中

PUBは、ブラザープリンターご愛用者のための「ユーザー友の会」。プリンターをサポートしたプログラムの募集・紹介・及びプリンターに関するハード・ソフト情報のコミュニケーションが主な活動内容です。入会者にはPUB会員証を進呈します。詳しくは(03)274-6911へどうぞ。

PUB MEDIA 60年秋号 好評配布中

PUBの会員誌PUB MEDIAの新しい号が発刊されます。M-1024の詳しい紹介記事も載っています。ご希望の方は、上記のPUB MEDIA編集部へお問い合わせください。

ブラザー販売株式会社 情報機器事業部

東京／〒104 東京都中央区京橋3-3-8 ☎(03)274-6911
名古屋／〒460 名古屋市中区大須3-46-15 ☎(052)263-5811
大阪／〒542 大阪市南区心斉橋筋1-1 ☎(06)251-7265

ブラザープリンターの詳しい資料をご希望の方は、右の番号のいずれかに○をつけ、はがきにこの部分を貼ってお送りください。また、お手持ちのパソコン機種、使用用途(ゲーム、ビジネス…など)、住所、お名前、年令、電話番号もお書きください。

1＝M-1024P/X
2＝HR-5/5X
3＝M-1009/9X
4＝HR-6X

資料請求券 oh! Hit Bit 1月号

Carry lab.

藤谷　美和君・20歳　JET-X1

熱烈パソコンワープロ

上司も
彼女も
大感激

強い味方は僕のもの

ヤア！ 僕、藤谷美和。大学生を3年間やっているハタチのいい男です。ミワではなくヨシカズと申します。身長177cm、体重84kg、ちょっと太めだけど自分の体型には満足しています。頑丈な体を活かしてグルメ同好会の会長をしていますが、本職はお星さまのお勉強をすることです。今話題のハレーすい星の軌道計算なんかもやるんですよ!! すごいでしょ。

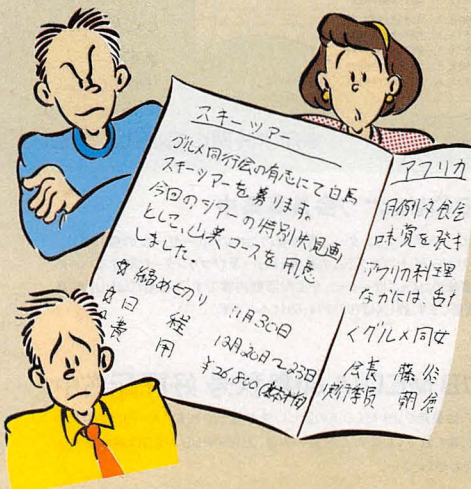
ところで、グルメ同好会といっても、ただ食べ歩くだけではありません。春はテニス、夏はサーフィン、秋はハイキング、冬はスキーと、スポーツをこよなく愛好しています。催し物の好きな連中が集っていますから、月1回の会報のほかに、案内状など文字との付き合いもけっこうなものなのです。

そして、悪筆で名高い僕は、なんとこの会報作りの功績によって―じつは JET-X1 のおかげなのですが― 現在まで会長として君臨しているのであります。

悪筆を自覚する時、そこにワープロがあった

春の会員総会、といってもメンバーが15人のグルメ同好会ですが、こんな提案がでました。

「次期会長になる人は、我が会念願の会報を出すことにしましょう。どなたか立候補しませんか」

そこで、会報を書くだけで会長になれるなら……と思った僕は、悪筆のことをすっかり忘れて手を挙げてしまったのでした。当時は僕の字体を知らない会員達は、体格に感心したのか、会報を書くという面倒を押しつけて、ほっとするためか、全員一致で僕を会長に承認してしまったのです。そして会報第1号が発行されました。

お世辞にも美しいとは言えないと自分でも感じてはいます。それでも第3号までは何の苦情も出ずに済み、そして臨時総会。

「会報はもっと品位のあるものにして下さい。せっかくの面白い情報も誤字、脱字、それに解読不可能な字が多くて読めません。」

―会長は何とかせよ!!―と強いお言葉が矢のように飛び交います。その時、矢の一本が頭に当り、ひらめくものがありました。(そうだ！ワープロがあるじゃないか。)

「皆さん、次の会報に期待して下さい。私は会長です。」

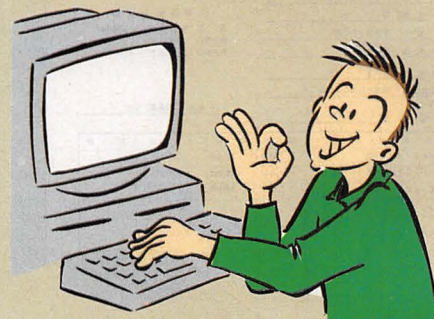
計算機をワープロに

僕は、本職がお星様の軌道計算なので、自宅にもX1turboを置いています。でも、会報事件が起きるまではパソコンには計算ばかりさせていました。根が凝性なのでワープロならなんでも良いという訳にはいきません。パソコンショップを数店歩き回り、チェックポイントをメモに抜き出しました。

- 入力から印刷までの操作は簡単か？
- 編集機能は充実しているか？
- 漢字変換機能と辞書内容は強力か？
- 表現力は豊富か？
- メンテナンス等、サポート体制はしっかりしているか？

そしてもう1つ、プリンタとの相性があります。ソフトによってはプリンタが限定されてしまい、確認せずに購入したため、コントロールコードの違いからうまく動作しないということも起こります。ワープロソフトの盲点でもありますから、購入するまえに十分確認しておきましょう。

JET-X1は、なぜ選ばれたか

僕の持っている X1/X1turbo 用のワープロとして数種類のソフトのデモを見ました。その結果、僕は JET-X1 に決めました。

使用目的の第1が会報作成ですから、表現力がなければダメです。JET-X1は10種類のアンダーライン、26種類の網かけ、そして倍角、¼角、半角、イタリック体と文字の変化にも富ます。それに本職の方では、計算式を扱いますから、¼角文字は大いに役立ちま

日本語ワープロJETシリーズ

JET-8801A
- PC-8801/mkII/SR/TR/FR/MR
- 5インチ(2D)3枚組 ¥35,800

Miss ワープロ
- PC-6601SR/6001mkIISR
- 3.5インチ(2枚組) ¥29,000

JET-8001SR
- PC-8001mkIISR
- 5インチ(2D)3枚組 ¥29,000

★FM-7/77シリーズ、MZ-2200/2000シリーズも、好評発売中
(仕様が異なりますので詳しくは、お問い合わせ下さい)

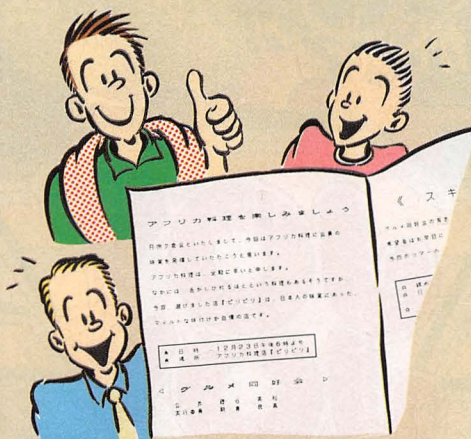
で、グルメ同好会の会長に納まるの巻

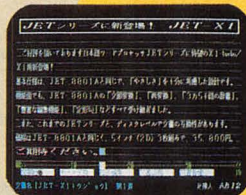
す。外字パターンを組み合わせればグルメ同好会のロゴでも作れるから楽しくなってしまいます。

そして最大のメリット、それは、わからないことが起きたとき、気軽に相談できるサポートセンターがあり、メンテナンスもバッチリの体制がととのっていることです。

これだけのパフォーマンスのあるソフトを35,800円で買えるのは、僕のような学生には嬉しい話しです。

JET-X1おおいに活躍!

僕のお気に入りのJET-X1は、操作が画面インフォメーションに従うだけで良いので、使い始めてすぐに全機能を引き出せます。公約通りに上出来の会報を出すことができ、会長としての腕前も磨きがかかってきました。どうです、グルメ同好会の品位が表現されているでしょう。

次回は、僕の本職(学業)である論文作成に、JETが役に立った話しをしましょう。

日本語ワープロJETシリーズ
販売実績 **30,000本達成!!**

JET-X1画面
●写真は、X1turboです

JET-X1
価格 ¥35,800
SHARP X1シリーズ/X1turboシリーズ
5インチ (2D) 3枚組

JET-X1 新発売
X1turboII も同時新発売!

●JET-X1仕様一覧 (X1シリーズとX1turboシリーズは、同一パッケージで販売しております)

項目	内容
文書サイズ	最大64文字×96行、一枚の文書ディスクに64文書まで記録
画面表示	40文字×9行(X1turbo 200ライン、X1シリーズ)。40文字×16行(X1turbo 400ラインのみ)。40文字×21行(X1turbo 400ラインのみ)
文字種類	全角(JIS第一水準漢字2965文字、非漢字403文字)。システム外字(全角、斜体字、記号など321文字)。ユーザー外字(全角、ユーザー定義可能 192文字)。半角文字(英・数・記号など96文字)。1/4角文字(数・かっこ記号など上つき下つき各16文字)。上記の倍角、縦倍角、4倍角(縦倍角、4倍角はプリンタ印字のみ)。JIS第二水準は漢字コード表示(プリンタにより第二水準印字可)。(注)縦倍角、4倍角はプリンタによっては印字できないものがあります
文字入力	かな入力、ローマ字入力、コード入力(JIS第二水準)。文字記号入力(記述記号、括弧記号、学術記号、単位記号、一般記号、数値記号、略字記号、ギリシャ文字、外字文字から選択、漢字コード)
文節変換	JET-CORE™搭載、文法解析による最適候補選択。接頭語、接尾語付文節変換、重変換、再変換、漢数字変換。辞書学習機能あり
編集	レイアウト機能、作表、アンダーライン(10種類)網かけ(26種類)、文字そろえ、ブロック複写、タブ設定、センタリング、右よせ、左よせ、文字列置換、文書復活(turboのみ)。行の挿入、削除、桁の挿入、削除、行の複写、文字の切り貼り(40文字まで)
辞書	約3万5千語を標準装備(一般、人名、地名を含む)、辞書学習機能ありユーザーによる辞書登録・削除可能、約4万語まで拡張可能
定型句	読みを含み64文字×96句登録可能。読み、またはファンクションキーで選択
印刷	各種プリンタに対応、縦書き 横書き印刷、行間 文字間設定、左マージン設定、印刷枚数、部分印刷、連続印刷など (注)縦倍角、4倍角の印字について ●8ピン・16ピンのドット・漢字プリンタでは可能 ●24ドット漢字プリンタはプリンタ本体に縦倍、4倍印字機能があれば可能(CZ-8PN1、MZ-1P17、PC-PR201H/Tなど)
補助機能	システム設定、辞書 文書コピー、全文書印刷、辞書管理、外字管理、文字列検索、文書変換(文書 BASICファイル)

●日本語処理システム JET-CORE™

「日本語ワープロJET-8801A」の日本語処理(かな漢字変換)部分を再構成した「日本語処理システムJET-CORE™」をOEM供給いたします。詳細につきましては、書面またはファックスにて当社企画部CORE(コア)係までご連絡ください 注)一般のお客様には市販いたしませんのでご了承ください

㈱キャリー・ラボ
〒862 熊本市大江6丁目25-25 金子ビル1F
TEL 096-363-0211 FAX 096-363-0235-G2・G3
サポートセンター(月〜金曜日14:00〜17:00)
電話にて受付 096-363-0047

(※お詫び)JET-X1の発売が遅れたことをお詫び申し上げます。

資料請求券 Oh!MZ 1月号

ソフトとボードゲームがドッキング 愉快 痛快 ジョイジョイ・パックⅢ

JOY JOY PACK SPECIAL

LICENSED FROM ©BRODERBUND SOFTWARE

ソフトゲームのスリルにボードゲームのワイワイ気分が加わってたのしさ新次元!!

JOYJOYPACK SPECIAL。噂のコンピュータゲーム「ドロール」とオリジナルの ボードゲームがドッキングした。ちょっぴりロマンチックなスペースウォーズはピンチありチャンスありの超大規模な構想。そのスペクタクルを秘中の作戦で切り拓くのは君か、それとも…。 狙いが当たれば一発逆転。みんなが集まってワイワイ盛り上がって楽しめるぞ。

GAME STORY

緊急指令//家族救出大作戦

★★★★ 地球編 ★★★★

宇宙大魔人に家族をさらわれたドロール君。救出作戦開始です。でも宇宙に旅立つためには莫大なお金が必要。困ったドロール君は資金作りのアルバイトを始めます。失敗を繰り返したり、宇宙大魔人にねらわれたり、思わぬ幸運にであったり。そしてついに出発準備完了。興奮に包まれながら宇宙大魔人の惑星に向けて、さぁ、スタート。

宇宙・救出編

大宇宙に広がるスペースファンタジー
宇宙に無事飛び出したドロール君、いろいろな人に助けられながら着々と戦闘準備を整えます。でも広い宇宙は危険もいっぱい。悪徳商人、宇宙海賊、魔のブラックホールと息もつかせぬピンチの連続です。ワープチャンスで一気に宇宙大魔人の惑星へ突撃。この最後の決戦がドロールゲームだ。救出なるか?!

勝負は最後までわからない・マネーウォーズ

JOYJOYPACK SPECIALのもうひとつの面白さは、ドロール君のマネーゲーム。ドロール君の貯金箱にはたったの$1000、でも宇宙船は$10000。アルバイト選びも楽じゃない、ウエイターに道路工事に新聞配達、うれしい給料日に宝くじの夢。宇宙へ旅だっても武器を買ったり宇宙船が故障したり。無事に家族を救出すれば、なんと$150000のボーナス。持ち金がモノをいって、ゲーム終了と同時アッと驚く大逆転！

GAME — DROL

QDボックス只今到着！

©BRODERBUND SOFTWARE

特殊スーツに身を包んだドロール君が、4階建ての地下牢を舞台にくりひろげるスリル満点のアクションゲーム!! 1面は妹とトカゲ、次に弟とワニ、最後にお母さんを助け出せば1セット終わり。感動的なデモが君を暖かく包んでくれるよ。サソリやエイリアン、へび、おまけにオノやナイフや鬼までも…。画面が進むごとにユニークなキャラクターが登場し、楽しさとスリルがいっぱい!! ドロール君はみんなを助けることができるかなー？

GAME — SHOOTING WARS

JOY JOY PACK SPECIAL用に新登場!!
君は、宇宙のスーパーヒーロー、敵の攻撃をかわしながら敵を撃つスリル満点のシューティングゲーム。また、ボードゲームとドッキングさせ違った魅力もいっぱい!! カード に(ゲームボード中)書いてあるアルファベットを簡単に入力するだけで、宇宙船の攻撃、防御、スピードなどいろいろ変わってたのしさ2倍、3倍…。さあ！君も宇宙バトルへ出発だ!!

●写真は実際の画面と多少異なる場合もあります。

ドロール君のメッセージ

僕がドロール。噂はもう聞いているね。アメリカ生まれの冒険少年。愉快な活躍で人気者なんだ。JOYJOYPACK SPECIALで、いよいよ君と一緒にスペースバトルの旅ができるヨ。今から楽しみなのさ。君の持っているMZ-1500で思う存分プレイすれば、もう夢中になれることマチガイナシ。

MZ-1500 QD 標準価格9,800円

当社では、Super MZ用のソフトウェアの開発、流通を展開しています。今年11月より専門のアプリケーション・ニュース誌「トラン」を創刊し、製品の紹介とあわせてSuper MZユーザー のあらゆる面でのサポートを行なってまいります。お問い合せは下記住所迄。

sbc エス・ビー・シーソフトウェア株式会社
〒160東京都新宿区本塩21木田建設ビル TEL.03-353-9241(大代)

スーパーMZやターボの馬力を120%活かした

ビジレス

●33本の応用実例集付
●電子マニュアル付
●電子コマンド辞書付
(X1ターボのみ)

カルク、スプレッドシート、リレーショナルデータベースのすべてがここにある。
シャープ8ビットマシンソフトのベストセラー、
超高速マシン語「ビジレス」。
33本のサンプル例を搭載して好評発売中。

ホビー、ファミリー、ビジネス…これ1本でパーフェクト。

「ビジネスは変更の連続」とソクラテスは定義した。そんなニーズにぴったりのマルチタスクワーカーがビジレスだ。

「1本のソフトで全て処理。
こんなソフトがあったらいいな。」
プログラムによってディスクの入れ替えはもういやだ。
住所録をせっかく作成したのにDMを発行しようと思ったらそのソフトに機能がないため再度DM用ソフトに入力しなおしなんて……顧客管理をしようと思ってもそのソフトでは機能が不足するために、また別のソフトを購入してしまう。また始めから入力をする……すると社長からせっかくコンピュータがあるのだから顧客の売上や仕入れを追加してABC分析をするように命令される。やっと入力が終了してほっとしていると、また社長から追加の言葉……今度はセールスマン別に分析しろとのこと……さらに担当セールスマン売上と顧客の仕入れを比較するべきとのこと……給与計算と顧客管理を連結せよなどとは社長も思いつくことはないだろうと思っていたら……恐れていた事態が発生した。やはりセールスマン管理で売上別分析ができたのだから、そのデータを利用して給与計算のなかに売上歩合も追加するとのこと……もう我慢できない……そんなことは不可能だ……助けてくれ……僕はプログラマーではない。
皆さんもそんな経験はありませんか?
ビジレスはそのようなニーズから誕生しました。

パーソナルビジレス版　定価28,000円

ビジレスの特徴

●80のコマンドもカンタン操作で、自由な表づくり
よこ130文字26項目以内、たて999行以内で使用。ケタ数と数値データ#か文字列$かの指定をおこなうだけで表のわくがつくれます。(カセット、QD版タテ256)

●必要なデータを項目別にすばやく探す検索機能
検索スピードは、250個のデータから0.5秒。記憶されている膨大なデータの中から条件にあてはまるデータを高速検索。データの分類も簡単。

●列単位セル間でウルトラ演算
表の列と列、定数の列の間で、加減乗除、%計算、構成比率、残高、累計、平均、最大、最小、標準偏差の計算ができます。

●情報ごとの集計が可能で、仕訳もラクラク
タテの合計、ヨコの合計、同じ項目名による集計は簡単です。またディスクにある別々のファイルの集計ももちろん可能です。

●よく使用する手順は自動プログラム
定期的な仕事を設定すれば、専用プログラム。ファイルごとにちがう手順も設定できます。
マルチファイル15,000文字以上(カセット82文字)

●シャープユーザー辞書でダブルデータベースになる
例:ユーザー辞書に郵便番号、コード等を登録、スピーディーに住所、名前を呼び出しビジレスに入力できる。

●1ヵ所変更ですべて変更できるデータ・リンク
一つの変更リストをもとに関連するファイル更新ができます。

●各種伝票のプリント・アウト。宛名印刷も可能
各種伝票や集計表、見積書はきれいなフォームでプリント・アウト。画面上で帳簿を設計することができ、宛名の印刷もこなします。

●データの並び替え可能
250個のデータを36秒でソート。データを大きい順、小さい順、あいうえお順に並び替えます。順位づけやデータの整理にたいへん有効。

リレーショナル・データベースとしてのビジレス利用の実際例

売上ファイル　昭和60年 1月 15日

年月日	品番	商品名	売上高	売先
59.10.01	H-100	ﾙﾋﾞﾅｽ	3,000	33
59.10.03	H-200	ﾏｰｶﾞﾚｯﾄ	8,000	12
59.10.03	H-300	ﾃﾞｰｼﾞｰ	5,000	33
59.10.05	H-500	ｼﾞｬｽﾐﾝ	7,500	15
59.10.05	H-200	ﾏｰｶﾞﾚｯﾄ	2,000	33
59.10.07	H-700	ｽｽﾞﾗﾝ	10,000	33
59.10.07	H-500	ｼﾞｬｽﾐﾝ	15,000	11
59.10.10	H-100	ﾙﾋﾞﾅｽ	7,000	12
59.10.12	H-500	ｼﾞｬｽﾐﾝ	10,000	33
合計			67,500	

プロジェクション >@ 集計

売上集計　60年 1月 15日

品番	商品名	売上高
H-100	ﾙﾋﾞﾅｽ	10,000
H-200	ﾏｰｶﾞﾚｯﾄ	10,000
H-300	ﾃﾞｰｼﾞｰ	5,000
H-500	ｼﾞｬｽﾐﾝ	32,500
H-700	ｽｽﾞﾗﾝ	10,000
合計		67,500

セレクション >B 分類

早川様お買上明細　昭和60年 1月 15日

年月日	品番	商品名	売上高	売先
59.10.01	H-100	ﾙﾋﾞﾅｽ	3,000	33
59.10.03	H-300	ﾃﾞｰｼﾞｰ	5,000	33
59.10.05	H-200	ﾏｰｶﾞﾚｯﾄ	2,000	33
59.10.07	H-700	ｽｽﾞﾗﾝ	10,000	33
59.10.12	H-500	ｼﾞｬｽﾐﾝ	10,000	33
合計			30,000	

得意先一覧表

売先コード	得意先名
11	山本商会
12	(株)ハヤシ
15	高木商店
33	早川商事

分析表　MASH

売先	得意先名	売上高	比率
33	早川商事	30,000	44.4%
12	(株)ハヤシ	15,000	22.2%
11	山本商会	15,000	22.2%
15	高木商店	7,500	11.2%
合計		67,500	100.0%

売上分析グラフ　MASH

得意先名	売上高	比率	....5....10...15...20...25...30..千円
早川商事	30,000	44.4%	000000000000000000000000000000
(株)ハヤシ	15,000	22.2%	000000000000000
山本商会	15,000	22.2%	000000000000000
高木商店	7,500	11.2%	0000000
合計	67,500	100.0%	

サンプル版でお試しください!

現金書留に3,000円を同封の上、住所・氏名・TEL・商品名・ご使用機種名をご記入の上、OAテック宛お申し込みください。

カセット版好評発売中!!

■対応機種
X1/F/ターボ/ターボII、MZ-2500/2200/2000/80B
※X1FはFD1台で使用可能。

■対応プリンタ
PC-PR201/PC-8822/NM-9400/UP-130K/CZ-8PN1/CZ-8PD2/CZ-80PK/CZ-800P/MZ-1P07/MZ-1P10/MZ-1P11

■価格
●TAPE版¥10,000●5インチ、3インチ版(カナ)¥39,000●5インチターボ漢字版¥48,000
●MZ-2200QD版¥12,000●3.5インチMZ-2500漢字版¥48,000(予価)

OAテック推薦強力"Let's シリーズ"

■下記ソフトを起動するのに「ビジレス」は不要です。■データは全て「ビジレス」で使用できます。■Let'sシリーズのデータはお互いにコンパチです。

●家計簿(¥6,800)●住所録/DM(¥15,000)●データベースワープロ(¥18,000)●レコード・ビデオ管理(¥15,000)●名刺管理(¥6,800)●ファミリーパック(10本入)(¥30,000)●カロリー計算(¥6,800)●献立サーチャー(¥6,800)●テスト成績処理(¥7,200)●生徒台帳(¥6,800)●成績記録簿(¥6,800)●塾月謝管理(¥6,800)●売上管理(¥18,000)●仕入管理(¥18,000)●売上/仕入管理(¥32,000)●マルチ表集計(¥15,000)●在庫管理(¥28,000)●顧客管理(¥15,000)●見積書発行(¥15,000)●納品/請求書発行(¥18,000)●レンタル物件管理(¥24,000)●金種計算(¥6,800)●アマチュア無線電子ログ(¥6,800)●ビジレス学習用ソフト(¥3,000)●営業マン成績管理(¥6,800)●セールスマン顧客管理(¥7,200)●会員管理(¥6,800)●パーソナルパック(10本入)(¥30,000)●ソフト整理簿(¥6,800)●簡易給与計算(¥15,000)●手形管理(¥6,800)

★32,000語熟語変換可(人名、地名)。シャープのユーザー辞書が使えます。

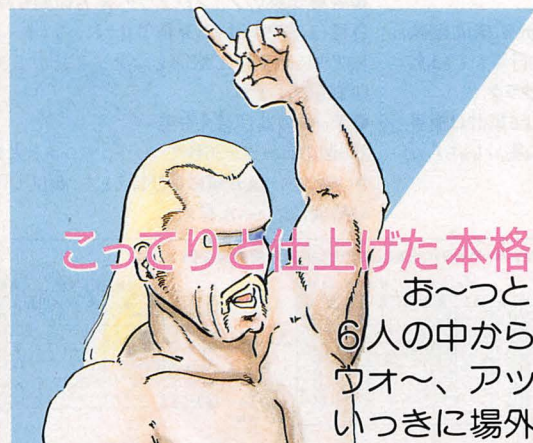
Personal Computer Programing Specialist

総発売元 ㈱OAテック

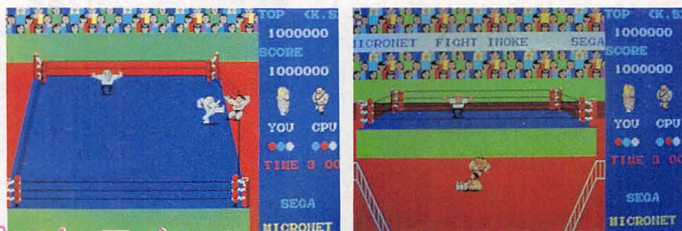
本社●〒444 愛知県岡崎市上六名3-13-2 ☎(0564)53-9400
●ビジレス開発元:システムデザイナーMASH 豊田市若宮町5-47☎(0565)31-7644
■最寄りのショップで購入できない場合はお電話でお申し込みください。■岡崎信用金庫六名支店 ㊟No.042283・OAテック■ソフトカタログをご希望される場合は、住所・氏名・年齢・ご使用機種名をご明記の上、資料請求券を同封してご請求ください。
■詳しいカタログをご希望の方は「解体新書」をお求めください。(1,500円分の切手を同封の上、OAテックまでご送付ください。)

資料請求券
oh! MZ1月号

フリッキーは青い鳥。のら猫ニャンニャンの手からひよこを守れ! アクションパズルゲーム!
全48面以上、FM音原、PCG、コンストラクションなど、それぞれの機種の個性を生かし、オリジナルをこえた!

産地 新発売 直送

こってりと仕上げた本格プロレス!

お〜っと、ついに登場チャンピオン・プロレススペシャル!
6人の中からレスラーをえらび必殺ワザをきめろ!
ウオ〜、アックスボンバー! コノヤロ、エンズイ切り!!
いっきに場外乱闘もOK!

CHAMPION
©SEGA プロレス Special

EXPRESSION OF SENSIBILITY & COMMUNICATION
Super MZ-2500 対応
チャンピオンプロレススペシャル&フリッキー

CT…カセット版　FD…フロッピー版　RC…ロム版　※X1-Dは使用不可

機種		価格
PC-8001 mkII/SR	CT	¥4,800
PC-8801/mkII/SR	CT	¥4,800
	FD	¥6,800
MZ-2000/2200/2500	CT	¥4,800
SHARPX1/turbo	※ CT	¥4,800
	※ FD	¥6,800
FM-7/NEW7	CT	¥4,800
	FD	¥6,800
MSX(フリッキーのみ)	RC	¥4,400

※この商品はセガ・エンタープライゼス社の許諾を受けて開発したものです。

マイクロネット
Micronet

〒064 札幌市中央区南10条西15丁目ムラカミビル3F ☎011-561-1370

RIGLAS
生
若き者よ、己を磨く剣を持て。語りかける愛を持て、
そして何よりも、清浄なる魂を持て。——
誕
好評発売中!!
FRPG ——— リグラス
X1 / X1 turbo
5インチ[2D] 定価6,800円
カセット発売予定
今までのゲームにない、ヒューマニズムにあふれたシナリオコンセプト。息をもつかせぬ、冒険また冒険の連続。あの俊才森田和郎率いる超絶プログラマー軍団『ランダムハウス』の自信作。いまベールを脱ぐ、壮大無比、心臓乱舞のファンタジーロマン。
森田和郎総監督,
シナリオ山口祐平
RIGLAS
SCORE
LIFE
FIGHT
主題性!集大成!したいぜ!
RANDOM HOUSE
ソフトウェア開発
株式会社ランダムハウス
埼玉県坂戸市末広町3-11
(営業所)TEL0298-42-1307

爆走バギー　エンジン全開、注目のスタート!!

一発野郎

グラフィックはリアルに3次元的処理
ステージは4セクション
軽快なサウンドにのって
君は5つのチェックポイントを通過できるか!?

●X-1/turbo/F　(CT)￥4,200 (5D)￥6,200
※DISK版はマップコントラクション付。
※X-1Dをご使用の方には当社で3インチ版のサポートをいたします。当社までご連絡下さい。

Super mz
mz-2500, X1シリーズ 堂々登場!

RELICS

レリクス

近日発売!

緊急速報! X1シリーズ 発売開始!

HOTDOG

ホットドッグ

超時空要塞 マクロス MACROSS
マクロスカウントダウン

◀おかげさまで好評発売中!

©ビックウエスト

★当社製品内にあるアンケートハガキにお答えの上、「カレンダーおくれ」と明記してお送りいただいた方の中から抽選で2000名様にボーステックオリジナルカレンダー('86年版)をプレゼント!!

応援します

® ボーステック株式会社
〒150 渋谷区神宮前5-42-1
TEL.(03)407-4191

★当社製品の開発スタッフを求めています。(自宅開発可)

●通信販売も行なっております。ご注文の際は、品名・機種名・住所・氏名・電話番号を明記の上、必ず現金書留でお申込下さい。(送料無料) 尚、お急ぎの方は、速達代金300円追加して下さい。

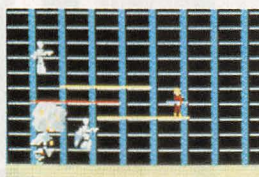
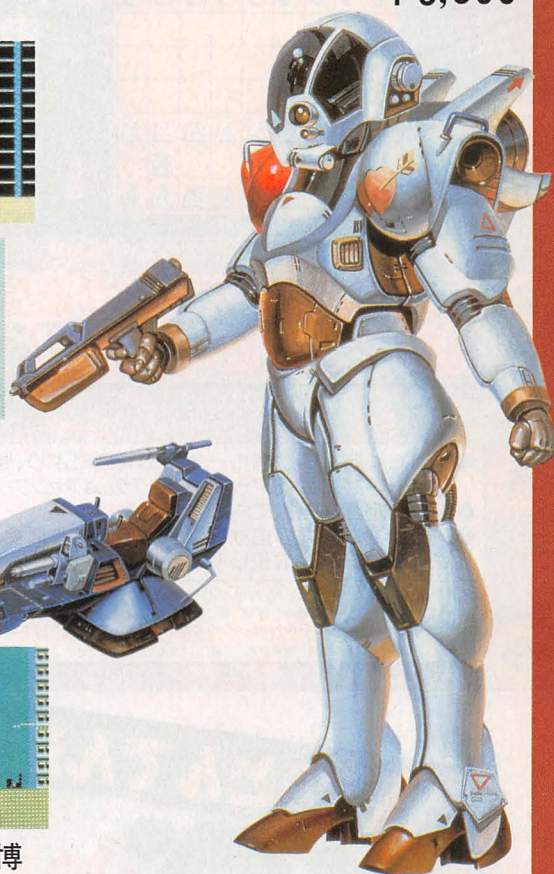
ブレインブレイカー

スーパーリアリズムロールプレイング

★母船からのメッセージで展開する
新感覚ロールプレイングアドベンチャー★

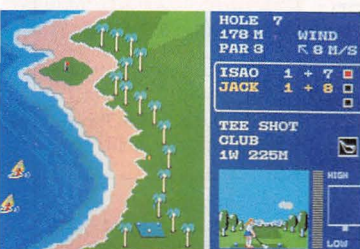
■本格的SFストーリーで展開するスーパーリアリズムロールプレイングゲーム。
■高速8方向スクロールでスムーズなゲーム展開。
■地上戦、空中戦、海戦と広大なマップに潜む無限の敵キャラクタ。
■上・下・屈伸、ジャンプ等、24パターンの動きをするリアルなキャラクタ。
■状況によって、画面下にメッセージが流れる新手法を採用。アドベンチャー的ムードもタップリ。
■ビームガン、ジェットパック等の強力兵器を装備していく、キャラクタ。リアリティ抜群の成長度に興奮まちがいなし。
■デュラン・デュランの「ユニオン・オブ・ザ・スネーク」を挿入。名曲と共にスリリングなゲームを楽しもう！

12月中旬 新発売

シャープX1シリーズ
5インチディスク版(ターボ可)…¥5,600
テープ版……………¥3,800

作者　石川　博

ワールドゴルフ

ゴルフゲームの最高峰

エニックスオープントーナメントゴルフ

近日発売

シャープX1シリーズ
5インチディスク版(ターボ可)
¥5,800
テープ版(2本組)…¥4,800

君はエニックスオープンの覇者となれるか!?

作者　村守将志

★リアリズムを追求した本格的機能★

●トッププロ30人とのトーナメント方式を採用。各プロの成績が、1ホールごとに表示され、キミの順位はすぐにわかる。18ホール終了後ベスト10以内に入れば、コースの会員として認められ、エキストラホール(9ホール)の使用許可がおりる。
●ゴルフコースの微妙なアンジュレーション(凸凹)を完全シミュレート。
●ボールの弾道もリアルそのもの。トップスピン、バックスピン、スライス、フック等、自由自在。
●美しいグラフィックで描く世界の著名ゴルフコース。

■〔3.5インチディスク版〕FM77AV、FM-77……¥5,800
■〔5インチディスク版〕PC-8801mkIISR、PC-8801mkII、PC-8801……¥5,800
■〔テープ版(2本組)〕FM-7シリーズ、PC-8801mkIISR、PC-8801mkII、PC-8801……¥4,800

TOKYO ナンパストリート

★このリアルなかけひきが君をプレイボーイにする★

人工知能型シミュレーションゲーム

シャープX1シリーズ
5インチディスク版(ターボ可)¥6,400
テープ版(2本組)…………¥4,800

作者　関野ひかる

★限りなく人工知能に近づいた女の子の豊富なリアクション★
★ナンパのノウハウを完全シミュレート★
★特定の女の子とデートできるデートモード付★

■〔5インチディスク版〕PC-9801/E/F/M/VM/VF 2D、2DD…¥6,400　2HD…¥7,400
■〔5インチディスク版〕PC-8801/mkII/SR、FM-7、FM-NEW7……¥6,400
■〔テープ版(2本組)〕PC-8801/mkII/SR、FM-7全シリーズ……¥4,800

軽井沢誘拐案内

ロマンチック ミステリー アドベンチャー

ポートピア連続殺人事件の作者・堀井雄二が放つロマンチックミステリーツアー！

シャープX1シリーズ
5インチディスク版(ターボ可)
¥5,800
テープ版(2本組)…¥4,800

ひたすら娯楽に徹したミステリー大作
君は軽井沢に愛を見たかっ!?

■〔5インチディスク版〕FM-7、FM-NEW7、PC-8801mkIISR、PC-8801mkII、PC-8801……¥5,800
■〔テープ版(2本組)〕FM-7全シリーズ、PC-8801mkIISR、PC-8801mkII、PC-8801¥4,800

★膨大なセリフデータをもち、つぎつぎかわってゆく登場人物たちのリアクション！
★画面数約100枚＋スクロールマップ3エリア！
★楽ちんなワンキー入力で、ともかく、めちゃめちゃ楽しい娯楽大作なのだ！
★読みやすい、ひらがな・カタカナ混合文。
★主人公の名前はキミ。おもわずてれてしまうボクなのだ。
★なぎさのオマケ写真もついて全6章。あふ♡

通信販売の御案内

御注文は現金書留にて、商品名、機種名、住所、氏名、電話番号を明記の上、お申し込み下さい。(送料無料)
〒160 東京都新宿区西新宿7-1-8
㈱エニックス「通信販売」係

販売元 株式会社 小西六エニックス
発行元 株式会社 エニックス
〒160 東京都新宿区西新宿7丁目1番8号 TEL03-366-4345

あなたのパソコンが最強の将棋マシンに早変り!!

もちろん、名人戦の設定・再現、駒落ち対局などは自由自在!

●まったく新しい思考ルーチンの開発により、強さ・スピード共に大幅アップ。(プログラムはアセンブリ言語)
●自由に定跡を登録できる。(FD)
●マイコンが人間の指す手を覚えてあなたに合った思考ルーチンに成長していきます。(FD)

★PC8801のユーザーで、棋太平が作動しない方へ
PC-8801(旧タイプ)をご使用の方で、稀に動作しないという方がいらっしゃいますが、作動しない方は、ディップスイッチをNベーシックモードにして立ち上げてみて下さい。
また、純正以外のディスクドライブをご使用の方も、同様にして立ち上げてみてください。

		機種	メディア・価格	備考
トンキー	GS 011	X-I/turbo シリーズ	3FD ¥5,800	CZ-800は、要G-RAM カラーモニタ使用 フロッピーディスクドライブ並びにデータレコーダは、純正品のみ動作確認済み ジョイスティック対応
	GS 012	X-I/turbo シリーズ	5FD ¥5,800	
	GS 013	X-I/turbo シリーズ	CT ¥3,800	
	GS 014	MZ-2200/2000 シリーズ	5FD ¥5,800	カラーモニタ使用 フロッピーディスクドライブ並びにデータレコーダは、純正品のみ動作確認済み
	GS 015	MZ-2200/2000 シリーズ	CT ¥3,800	
	GS 017	PC-8801/mk2/SR シリーズ	5FD ¥5,800	カラーモニタ使用 フロッピーディスクドライブは、純正品のみ動作確認済み
ぐるっぺ	GS 031	X-I/turboシリーズ MZ-2200/2000シリーズ PC-8801/mk2/SRシリーズ	5FD ¥5,800	CZ-800は、要G-RAM MZ2000は、要G-RAM1.2.3 カラーモニタ使用 フロッピーディスクドライブは、純正品のみ動作確認済み X-1シリーズは、ジョイスティック対応
	GS 032	X-I/turboシリーズ MZ-2200/2000シリーズ	CT ¥3,800	CZ-800は、要G-RAM MZ-2000は、要G-RAM1.2.3 カラーモニター使用 データレコーダは純正品のみ動作確認済み X-1シリーズは、ジョイスティック対応
	GS 036	PC-8801/mk2/SR シリーズ	CT ¥3,800	カラーモニタ並びにデータレコーダは、純正品のみ動作確認済み
	GS 037	X-I/turbo シリーズ	3FD ¥5,800	カラーモニタ使用 フロッピーディスクドライブは、純正品のみ動作確認済 ジョイスティック対応
棋太平	GS 051	X-I/turbo シリーズ	5FD ¥6,500	CZ-800は、要G-RAM カラーモニタ使用 フロッピーディスクドライブ並びにデータレコーダは、純正品のみ動作確認済み ジョイスティック対応 純正マウス対応 (ただし、X-1Dのテープ版使用は、できません)
	GS 052	X-I/turbo シリーズ	CT ¥4,500	
	GS 053	MZ-2200/2000 シリーズ	5FD ¥6,500	MZ-2000は、要G-RAM 1.2.3 グリーンモニタ使用可 フロッピーディスクドライブ並びにデーターレコーダーは、純正品のみ動作確認済み
	GS 054	MZ-2200/2000 シリーズ	CT ¥4,500	
	GS 055	PC-8801/mk2/SR シリーズ	5FD ¥6,500	カラーモニタ使用 フロッピーディスクドライブ並びにデータレコーダは、純正品のみ動作確認済み アスキーマウス対応
	GS 056	PC-8801/mk2/SR シリーズ	CT ¥4,500	
	GS 057	MZ-2500	3.5FD ¥7,000	カラーモニタ使用 ジョイスティック対応 純正マウス対応

㈱マイコンハウス SPS
〒960 福島市太平寺字町の内5-3☎(0245)45-5777
FAX(0245)45-1804 (GII, GIII)

お求めはお近くの有名マイコンショップで。
通信販売をご希望のかたは、商品名、機種名、メディアを明記のうえ料金を現金書留で当社までお申し込みください。(送料サービス)

パートナーショップ
キャリーラボ　マイクロキャビン

CROSS MEDIA SOFT

敵はメチャクチャ強い。

ZION-1 HI-003700 SCORE-003760

▲敵基地では攻撃をうまくよけながら、ザイオンと合体せよ!

STAGE-3 ZION-1 HI-003700 SCORE-002230

◀敵の大型空母ヘルダイバーを追ってワープゾーンに突入!

西暦2500年。宇宙のはるかかなたから、ギラン星人がやってきた。彼らは地球を守る合体ロボ・ザイオンを奪うと、宇宙空間の基地へ運んでしまった。地球が危ない。敵基地へ乗りこんで、ザイオンを取り戻せ!キミのキーテクニックで、地球を救うのだ。

- ●ザイオンを救出して合体せよ!どんどん強くなっていくぞ。
- ●全8面がそれぞれ違ったゲーム仕立て。1つのソフトで8倍楽しめる。
- ●合体が完了すると、超大型ロボ・サターンとの一騎打ちだ!!

超時空合体スペースファイト

ZION

ザイオン

●発売中(一部近日発売)

X1/X1turbo　5インチFD/カセット

※すべて　ディスク版　¥6,800

カセット版　¥5,800

FMシリーズ、PCシリーズも好評発売中

ムチとローソクの使い方教えてあげる

——かぐや姫——

ガッコーやオフィスじゃ、ぜったい教えてくれないけど、コレッてとっても役に立つのよね。「新竹取物語」のスタートの場面で、この2つをもらえるから、あなたの知識と経験で、うまく使いこなしてみて。実は私、知ってるんです。コレのスゴイ使い方。ホントーは教えてあげたいんだけど……、私の口からは言えないっ。だってとっても恥ずかしいことなんですもの、アドベンチャーのヒントを人に教えるなんて。あなたの健全な心で想像してね。そして高得点で私を捜し出してよね。かぐや姫からのお願いよ♡。

●もう、キミはかぐや姫に出逢ったか!?

新竹取物語

X1ユーザーの皆様、ながらくお待たせしました。

●絶賛発売中

X1/X1turbo　5インチFD

※すべて　ディスク版のみ　¥9,800(2枚組)

FMシリーズ、PCシリーズも好評発売中

(かぐや姫の親切アドバイス)

- ●未亡人のセイコさんも逢いたがってるわ。
- ●ポイント制で、ゲームの出来を競えるの。
- ●100点以上で私と逢えたら、段位をさしあげます。(直接問い合わせてね)
- ●よーく考えないと、ゲイバーに売られちゃうんだから。
- ●醜女姫と出逢っても、くじけないでもう一度ガンバッテ!

通販　商品名・使用機種名記入の上、代金と送料(500円)を現金書留にて、下記まで直接お申し込みください。
〒107 東京都港区北青山3-6-18(共同ビル青山2F)
日本エイ・ブイ・シー株式会社Oh!MZ係

募集　販売店を募集しています。
お問い合わせは、日本エイ・ブイ・シー株式会社企画部まで。☎03(486)4121
★健康で明るく、集中力のあるプログラマー、企画力のあるプロデューサー募集!
ビクター音楽産業株式会社 PS制作部まで。☎03(486)9470

●販売 avc 日本エイ・ブイ・シー株式会社　●発売 ビクター音楽産業株式会社

※諸般の事情により、発売が遅れることがございます。申し訳ありませんが、ご了承ください。

未体験

コンピュート・ミステリー

第1弾

担当刑事の報告書、
現場写真　数枚、
証拠物件　数点、
犯罪捜査局の大型コンピュータに
アクセスできるプログラムディスク一枚。
これらを手掛りに、殺人事件を解決せよ。
名探偵殿

依頼人　ハドソン

「暗闇の視点」

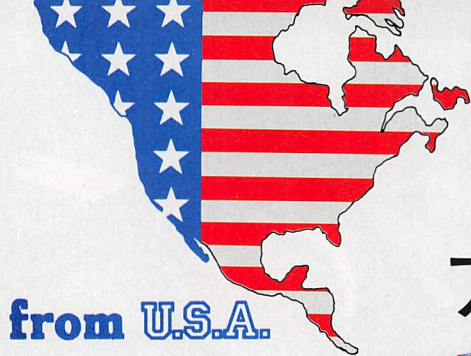
バニーガール　殺人事件。

バニーガール、小泉小百合が結婚することになり、店のスタッフやなじみ客が集まって、パーティが開催された。ところが、このパーティの最中に停電がおこり、小百合が何者かに殺害された。事件の解決が難航を極めた時、担当刑事はかつての先輩で現在探偵をしているX氏に、仕事を依頼した。

さて、あなたはこの探偵である。そして、与えられたデータとプログラムディスクを駆使し、犯人を推理せよ。

適応機種●PC-9801シリーズ●PC-8801mkII／SR●X1turbo〔FD版¥6,800〕

HUDSON GROUP
HUDSON SOFT®
本社・ハドソン札幌／〒062 札幌市豊平区平岸3条5丁目1-18
ハドソンビル PHONE:011-841-4622
営業所・東北・金沢・東京・名古屋・大阪・岡山・広島・福岡・鹿児島・沖縄
アメリカ・イギリス・西ドイツ

ウルトラ・アドベンチャー
『デゼニワールド』近日公開。
DEZENI WORLD
デゼニワールド
もっともっと面白いものを作ろうということで、発売予定が大幅に遅れていた『デゼニワールド』がついに完成した。（本当に、お待たせしてごめんなさい。）しかし、さすがに長い月日をかけたかいがあり、「これこそ、まさしくアドベンチャー・ゲームと呼ぶにふさわしい。」と天才竹・中コンビも自画自賛するほど素晴しい出来ばえとなった。しかも、嬉しいことにディスク版には、テーマソングなどが入ったサウンドトラックカセットも付いている。スリル！ 冒険！ 笑い！ ショッキングな旋風を巻き起こす『デゼニワールド』は、全国の家庭で近日公開。
適応機種 ●X-1 ●X-1C ●X-1turbo ●X-1F 〔5"FD版 ¥6,800〕
※画面の写真は全てX-1用で撮影したものです。
WARNING FOR CRAZY PEOPLE ONLY!
HUDSON GROUP
HUDSON SOFT®
本社・ハドソン札幌／〒062 札幌市豊平区平岸3条5丁目1-18
ハドソンビル PHONE:011-841-4622
営業所・東北・金沢・東京・名古屋・大阪・岡山・広島・福岡・鹿児島・沖縄
アメリカ・イギリス・西ドイツ
ハドソンの商品は、全国有名デパートおよびパソコンショップでお求めください

Licensed from
Brøderbund Software

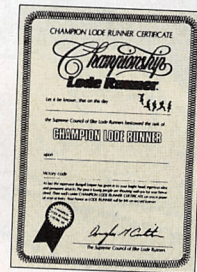
過激に遊ぼう!

アメリカは熱いゲームリングだ。

from U.S.A.

Championship Lode Runner™ チャンピオンシップロードランナー

X1シリーズ・FM-7シリーズ
テープ版 ¥4,800　フロッピィ版 ¥6,800

帝国の逆襲が始まった。新たに出現した要塞迷路は超難解，恐怖の50画面だ。君は再び挑戦する。君の頭脳を極限まで痛めつけるこの画面。果たして君は耐えられるか。全画面を駆けぬけたら，全米ロードランナー審議会の認定証をあげる!!

ロードランナー Lode Runner™

MZ-2500　フロッピィ版 ¥6,800
X1シリーズ・FM-7シリーズ・S1
テープ版 ¥4,800　フロッピィ版 ¥5,800
B16/EX、MX　IBMパーソナルコンピューターJX
フロッピィ版 ¥6,800

150もの迷路シーンがすごい。君は隠された黄金を求めて走る，走る！敵の手を逃れ，ハシゴを昇り，ジャンプする。君のオリジナルゲームも作れる，全米No.1ソフト，ロードランナー。

スペアチェンジ Spare Change™

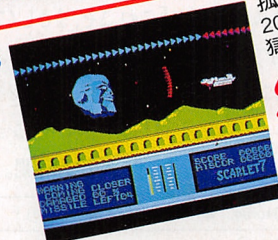
X1 FM-7シリーズ
テープ版 ¥4,800 フロッピィ版 ¥5,800

こいつは異変，大変だ。ゲームマシンからおかしなザークたちが飛びだして君のカジノを荒らし始めたぞ。君はあらゆるトリックを使って大切なカジノを守らなくてはならない！ゲームの難易度を自由に変えられるオモシロゲーム，スペアチェンジ。

チョップリフター CHOPLIFTER!

X1 FM-7シリーズ
テープ版 ¥4,800 フロッピィ版 ¥5,800

64人の捕虜全員を救出せよ。君は最新鋭ジェットヘリ“チョップリフター”を操り戦火に包まれた砂の帝国クリゴンへと向う。陸から空から激しい敵の攻撃。0.1秒を争う決死の救出作戦。君は空のヒーローになれるか。

Apollo Technica

孤立した工作部隊を救出せよ
2035年夏，スカーレット7は赤い地獄を見た。

スカーレット SCARLET7

X1シリーズ
テープ版 ¥3,800
フロッピィ版 ¥5,800

ソフトプロ株式会社
ソフトプロインターナショナル事業部　アポロテクニカ事業部
〒530　大阪市北区西天満6-7-2 梅新東ビル5F　TEL.06(363)1221

創刊1周年記念・特大号 1月号 特別定価390円

新春特別付録
BEEPカレンダー1986

特集I
ま、ファミコンなワケだ!!
ツインビー/マッハライダー/マクロス/スターラスターほか

特集II
名作ロールプレイングゲーム研究編
ウルティマII, III, IV
ウィザードリー

徹底研究
●スクリーマー

緊急レポート
●ブラスティー

コンピュータゲーム情報誌

好評発売中!

ファミリー コンピュータ
'86ゲームハンドブック
Beep別冊●定価800円●AB版144ページ

●ファミコン人気ゲームのテクやマップがバッチリ!
●キミが書きこむスコア表・マップ表もついている!
テグザー/カラテカ/キン肉マン/ポートピア連続殺人事件/ガルほか

ファミリー ベーシック
おもしろゲーム大作戦
BEEP編集●定価2,600円●B5判(本+カセットテープ)

●「マリオの走り幅とび」はじめ、おもしろゲーム19本!
●ファミリーベーシックの実用テクニックを紹介(機械語入門付)
●ファミリーベーシックの開発にたずさわったハドソン・竹部さん大いに語る。

(株)日本ソフトバンク ☎03・261・4095

Gallery by X1/X1turbo

カラーイメージボード

スペースキーをポン。まるでシャッターを押すようにテレビやビデオの画像が取り込める。カラーイメージボードは今いちばんウレシイ周辺機器だ。パソコンに落とした画像は自由に編集でき，嬉楽画に移したり，プリントアウトするためのソフトも万全。これはオシャレ！

400ラインフルカラーで富士山やオームさんもこんなに自然にCGになるのです。

コーヒーカップの輪郭抽出をしてみました。
新しいイメージがわいてきそうです。

嬉楽画やポップを使えばこんなユニークな
画像も自由に作れます。面白いでしょう。

カラーイメージボードの㊙技術解説

カラーイメージボードの特長を以下にまとめてみよう。

1)　X1シリーズ全機種で利用できる

I/Oスロットにボードを差し込み，コネクタで外部ユニットと接続するだけで，簡単にビデオ/テレビ映像のデジタイズができる。最近はFM77AVやMSX2のようにビジュアル関連に重点をおいたパソコンが注目を集めているが，X1/X1turboシリーズではカラーイメージボードを接続するだけでビデオデジタイズができるのだ。

2)　デジタイズが高速である

デジタイズ時間は1/60秒，画像取り込み時間はX1turboで約0.2秒，X1でも約1秒と超高速であるため，リアルタイムな画像処理が可能となる。

3)　中間色で画像取り込みができる

1)より，使用できる色は8色に限られるが，独自のスクランブル回路によってタイルパターンで画像を取り込むので，きわめて自然なカラーデジタイズができる。また，ハードコピーをとるのに特別なプリンタやツールを必要とせず，従来の周辺機器，グラフィックツールなどもそのまま使えるのだ。

それでは，カラーイメージボードによる画像取り込みの仕組みについて解説しよう。回路は大きく分けて3つの部分からなっている。外部ユニット内のデコーダ，スクランブル回路とボード内のバッファメモリである。

デコーダは入力端子から入ってきたビデオ信号（NTSCコンポジット）をデコードしてRGBのアナログ信号に変換する。スクランブル回路はその信号を“0”と“1”のデジタル信号に変換し，バッファメモリがそれを記憶する。パソコン側ではそのデータをI/Oポートから読み出してグラフィックメモリに転送するだけでよい。

スクランブル回路からの出力は，入力信号のRGBそれぞれについて，その階調に応じてパルス（“0”と“1”）の数と間隔が変化するようになっている。それはグラフィック画面のタイルパターンと同じであるため，きわめて自然な中間色表現ができるのだ。また，64Kバイトの容量を持つバッファメモリにはシリアルアクセス方式のダイナミックメモリが使われており，スクランブル信号をそのまま記憶することができるので，1/60秒という高速デジタイズが可能となるのである。

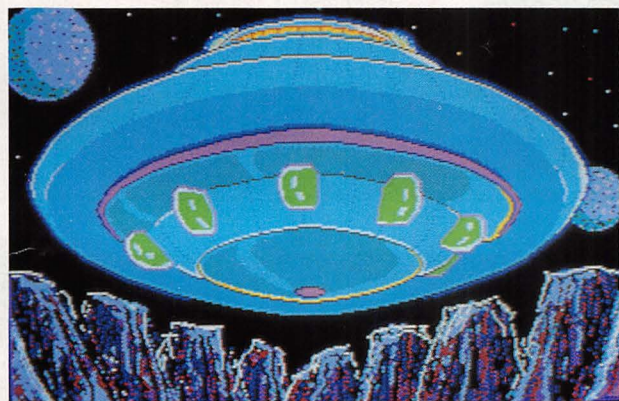
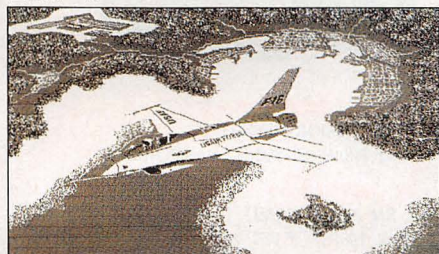
取り込み映像の調整は外部ユニットの映像調整ボリュームと色合い・色の濃さ・階調の3つのノブでできるが，ノブはドライバーで回す必要がある。取り込む「絵」によって細かく調整したくなるのがマニア心というものだ。もう少し扱いやすくしてほしかったように思う。

カラーイメージボードはスクランブル回路を始めとするユニークなアイデアにより，回路構成がたいへんシンプルになって初めて実現した，ローコストで信頼性の高いビデオデジタイズシステムである。　（中川　智哉）

「印刷ごっこ」でカラーハードコピー

「印刷ごっこ」はX1turboのBASICで作成したグラフィック図形を，漢字カラーサーマルプリンタMZ-1P17でハードコピーを取るためのツールです。付属のユーティリティを使えば，嬉楽画ターボで描いた絵でもご覧のように美しいハードコピーが可能です。

お馴染み嬉楽画turboのF-16

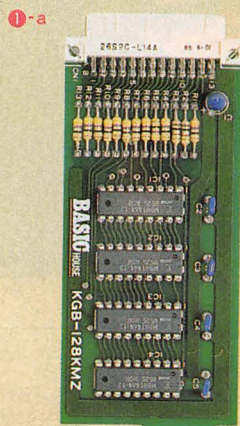
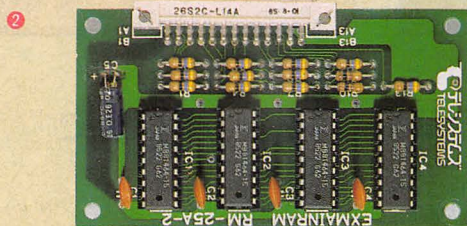
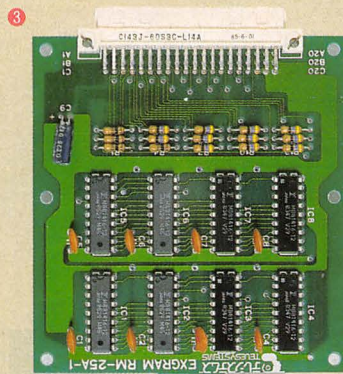
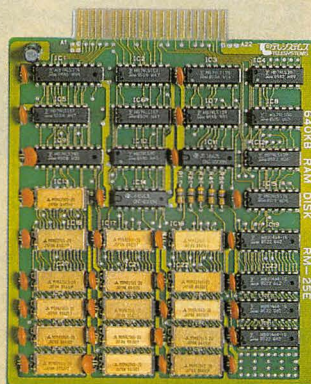
色分解の例

SUPER RAM BOARDでアニメーション

大容量1MバイトのSUPER RAM BOARD（デジック）にグラフィック画面を収めれば，本格的なアニメーショングラフィックスだってX1/X1turboにとって不可能ではありません。

ずらりと揃った大容量RAMボード

パソコンのメモリ拡張はとどまるところを知らずといった状況ですが，需要の増大と量産効果によってRAMはこんなに手に入りやすくなってきたのです。メインメモリやビデオRAMの増設から，大容量RAMディスクまで，使い方しだいであなたのパソコンは大幅にパワーアップです。

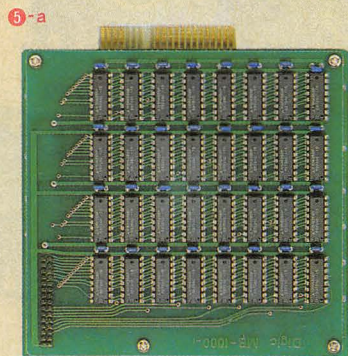
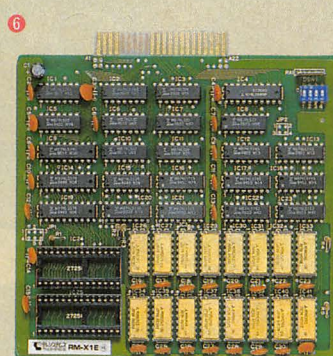
for MZ-2500 (→204ページ)

❶-a　❶-b　❷　❸　❹

❶-a, b　増設RAMボード
KGB-128KMZ(128KB)
12,800円→9,800円　BASIC HOUSE

❷　増設RAMカード
RM-25A-2(128KB)
12,100円　テレシステムズ

❸　増設ビデオRAMカード
RM-25A-1(64KB)
13,100円　テレシステムズ
(②と③のセットRM-25A　24,800円)

❹　RAM DISK
RM-25E(640KB)
49,800円　テレシステムズ

for X1/X1turbo (→204ページ)

❺-a, b　SUPER RAM BOARD
MB-1000(1MB)
(ユーティリティ付き)
99,800円　デジック

❻　RAM DISK
RM-X1E(512KB)
49,800円　テレシステムズ

LOGO ふたつの顔 第1回

タートルグラフィックスの世界

——カメと画面処理——

Mukouhara Ayumu
向原あゆむ

LOGOのX1 turbo版がついに登場です。タートルグラフィックスはもとより，X1 turboならではの日本語処理機能やリスト処理など，さまざまなプログラミング言語の側面を見せてくれることでしょう。これから面白い使い方が期待できそうですね。

この日をどんなに待ちわびたことでしょうか。X1 turboでもLOGO（ロゴと読みます）が動かせる日がやってきましたね。LOGOというプログラミング言語は，何か教育ママのようなウサンクサイ印象を持ってしまいますが，そんな堅苦しいことなど気にとめる必要はありません。私たちは，タートルグラフィックスを備えたリスト処理言語を楽しめばそれでよいのです。

それでは，LOGOとはどんな言語なのか見ていくことにしましょう。なお，話はturbo LOGOを中心に進めていきますが，LOGOの文法はどれも同じようなものですから，例題はほとんどそのまま他のLOGOでも動かせると思います。

1 リスト処理に進む前に

LOGOと言われて，少し常識ある人なら，「ああ，あのタートルグラフィックスのついた子供のための言語ですか」と答えることでしょう。確かに，こういう考え方は，LOGOの生まれた経緯から見れば非常に正しいことなのですが，LOGOのすべてではありません。タートルグラフィックスは，LOGOにとっては子供が自発的に学んでいくための手段，つまりは教育という営業用の顔なのです。しかし，本来の顔は，LOGOの兄貴にあたるLISPからほとんどそのまま受け継いできたリスト処理機能なのです。

LIPSは有名な人工知能用言語ですが，カッコをやたら使うので，プログラムが見にくく敬遠されがちです。しかし，LOGOではそのような欠点は多少修正されています。このため，見やすいLISPとしてLOGOを使用することもできるのです。ですから，LISPで行われている人工知能をもっと身近なものとして体験することも可能でしょう。

ところで，タートルグラフィックスとリスト処理という2つの側面は，LOGOではうまく一体化しているように見えますが，ここに落とし穴が隠されています。初心者はタートルグラフィックスに魅かれてLOGOを学び始め，グラフィックスに飽きたころ，リスト処理へ進もうとします。しかし，タートルグラフィックスを学ぶような感覚でリスト処理をやろうとしても，きっと挫折するに違いありません。そこには，はっきりとした気持ちの転換が必要なのです。考えてみれば，絵が上手に描けたからといって，リスト処理ができるようになるわけがないのです。したがって，LOGOに接するときには，自分の態度をハッキリさせておかねばなりません。自分はタートルグラフィックスをやりたいのか，リスト処理をやりたいのかということを。

```
X1 Turbo LOGO  Ver 1.00  Copyright (C) 1985 ACCESS CO.,LTD.
*** WELCOME TO FANTASIC LOGO ***
?> SHOWTURTLE
?>
```

図1 この三角形がカメだって？

それでは，これからLOGOのタートルグラフィックスについて説明していきたいと思います。これは次回に行う予定のリスト処理の前座みたいなものですから，気楽に読んでみてください。

2 カメとグラフィックス

これまで，何の気なしに「タートルグラフィックス」という言葉を使ってしまいましたが，これについて説明をまずしなければなりません。タートルとは海ガメのことです（ちなみに，陸ガメはトータス）が，このタートルを用いてお絵描きをしようというのが，タートルグラフィックスの世界なのです。

カメとグラフィックス。この奇妙な取り合わせはどのようなものなのか，これから説明していくことにしましょう。

それでは，LOGOを起動して，

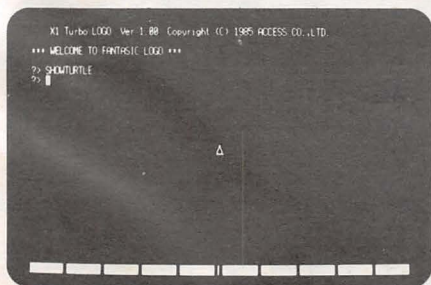
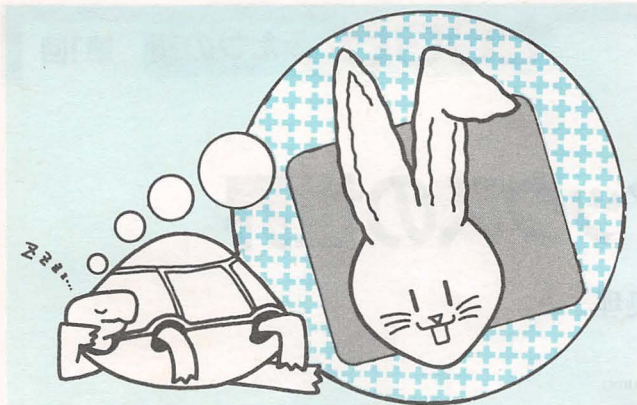
SHOWTURTLE

または，

ST

とキーボードから打ち込んでみてください。画面の中央に三角のしるしが現れてきます（図1）。何を隠そうこの三角形が，タートルグラフィックスで主役を演じるタートル君なのです。こんな三角形がカメだって，と驚くのはまだ早い。このタートル君はからだのどこかにペンを持っていて，自分の移動した跡に線を残していくのです。タートル君は，「前進しろ」とか，「後退しろ」とか，「右の方を向け」とか，「左の方を向け」

私たちがペンを持って紙の上に絵を描くのと同じような感覚ですね。

一般的には，絵を描くときに紙の上に座標軸を設定して，点（○○，□□）から，点(△△，××）まで線を引けるなんて考えている人はいないでしょう。このように考えると，タートルグラフィックスは，より自然な形で絵を描く方法だと言うことができます。これなら，座標系が何であるかを知らない小さな子供にも容易に理解することができますね（そもそも座標系とは，物の位置を把握するために人間が勝手に導入するものですから，算数なり数学なりで，そういうものがあることを教えられなくては理解することができないでしょう)。まあ，口上はこのくらいにして，実際にタートル君に動いてもらうことにしましょう。

といった人間からの指令を素直に実行します。このときに，タートル君の動いた跡が図形として残るというわけなのです。そして，これがタートルグラフィックス唯一の原理と言えます。これは，平面上に座標軸を設定して，x座標が○○でy座標が□□の点から，x座標が△△でy座標が××の点まで直線を引くという，BASICなどのグラフィックスとは根本的に異なっているのがわかるでしょう（実はLOGOでもx座標，y座標という概念はあるのですが，あまり強調されません)。たとえば，実際のプログラム(ひとつの命令？)と比べるとよくわかります。BASICで直線を引く場合のプログラムは，

LINE　(○○,□□)－(△△,××)

ですが，これがLOGOになると

FORWARD　☆☆

となります。FORWARDとはタートル君を前進させる命令で，上の例は

「☆☆という距離だけ前進しろ」

という命令，つまり，

「☆☆という長さの直線を引け」

という命令になります。ここには座標なんてものは出てきません。タートル君は，彼がいた場所から，彼が向いていた方向に☆☆という距離だけ進むだけです。これは，

ここでは，まず，FORWARD(省略形はFD）と，RIGHT（省略形はRT)を使用します。FORWARDは先程説明しましたが，RIGHT とはタートル君が現在向いている方向（三角形の頂点が向いている方向）より，与えられた角度だけ，さらに右を向けという命令です。ですから，

```
FORWARD 150    RIGHT 120
FORWARD 150    RIGHT 120
FORWARD 150    RIGHT 120
```

によって描かれる図形（図2）は一辺の長さが150である正三角形となります。ここで，最後の

RIGHT 120

は余分なような気がしますが，これはタートル君の状態を正三角形を描く前と同じ状態に戻すために付け加えたものです。絵を描く前後で，タートル君の状態（特に向いている方向）を変えないようにすることは，いろいろな図形を組み合わせて描かせるときに必要となってきますから，今のうちから習慣にしておきましょう。

ところで，前記の例ではまったく同じ命令を3回も入力しています。これは，あまりにも非効率的です。「以下の命令を3回繰り返せ」という命令があれば，すごく簡略化できます（正三角形ならまだよいのですが，正100角形とか，正360角形を描かせようとすると，前のやり方では死んでしまうでしょう)。もちろん，LOGOでも繰り返しのための命令があって，REPEATというのがそれです。REPEAT 命令は，

REPEAT ☆ [□ △ × ○ …]

という形式で使用します。このとき，[と]で囲まれた中が☆回繰り返されます。つまり，正三角形を描かせるためには，

REPEAT　3　[FD 150 RT 120]

と入力すればよかったのです（省略形を使ってしまった)。ちなみに，正100角形なら，

REPEAT 100 [FD 10 RT 3.6]

となり(図3)，正360角形なら，

REPEAT 360 [FD 2 RT 1]

となります(図4)。なお，これらの図形では，図形が画面からはみ出ないように前進させる距離を適当に変えてあります。また，右を向く角度は（子供に教えるときには，試行錯誤から発見させるのがLOGOのやり方ですが)，凸多角形の外角の和が360°であることを知っていれば簡単に求まるでしょう。

3 命令どおりに動いてね

LOGOの特徴のひとつに，新しい命令を作り出せるということがあります。これは，

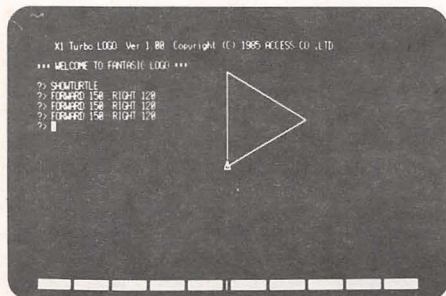



図2　三角形が三角形を描きました

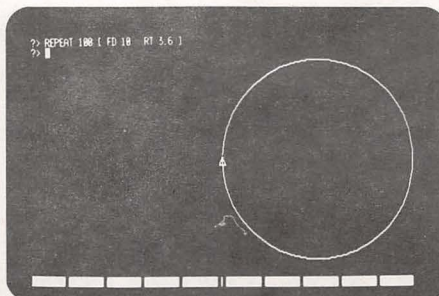



図3　正100角形はほとんど円みたい

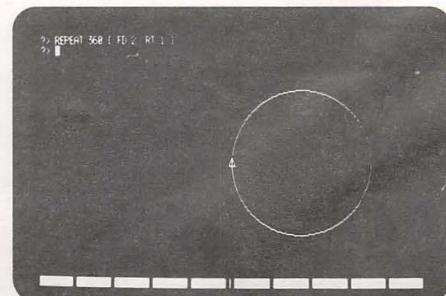



図4　正360角形も円と区別できない

リスト1

```
?>
?>
?> TO 正三角形
>> REPEAT 3 [ FD 150  RT 120 ]
>> END
正三角形 DEFINED
?>
?>
?>
```

リスト2

```
?>
?>
?> MAKE "REDEFP "TRUE
?>
?> TO 正三角形 :長さ
>> REPEAT 3 [ FD :長さ  RT 120 ]
>> END
正三角形 DEFINED
?>
?>
?>
?>
```

一連の操作に名前を付けることです。たとえば，先程の正三角形を描かせる命令，

REPEAT　3 ［ FD 150 RT 120 ］

に“正三角形”という名前を付けると，それからは，

正三角形

と入力するだけで，正三角形が描かれるようになります。このように，新しい命令を作るための命令がTOです。これは英語でいうところの不定詞，つまり，

TO　＋　動詞の原形

の形で用いられるTOのことです。これには，「～するためには」という意味がありますから，新しい命令を定義するにはぴったりな名前ですね（でも日本人にとっては関係ないか)。TOを使って“正三角形”という命令を定義するためには，

TO　正三角形

と入力します。すると，プロンプトが

？>

から

>>

に変わって命令の定義が可能であることを教えてくれます。ここで，

REPEAT　3 ［ FD 150 RT 120 ］

を入力したあと，

END

を入力して，定義の終了です(リスト1)。さあ，これでこの“正三角形”という命令は，タートルが初めから知っていた命令（こういう命令のことをLOGOではプリミティブと言います）と区別することなく使えるようになりました（これに対して，TOによって定義された命令はプロシージャと言います)。たとえば，普通の命令と同じように“正三角形”という命令を使って，

REPEAT　24 ［正三角形　RT　15 ］

という繰り返し命令を実行すれば，正三角形を回転させてできる図形を描かせることができるのです(図5)。これは，BASICでは見ることのできない特徴です（でも，LISPやFORTHでは決してめずらしいことではありません。知らんぷり，知らんぷり)。

ところで，前記のようにして定義した，“正三角形”という命令は，いつも一辺の長さが150の正三角形しか描くことができないので，おもしろくありません。FORWARDなどのプリミティブのように，命令に与える引数で一辺の長さを指定したいというのが人情でしょう。つまり，

正三角形　50

なら，一辺の長さが50の正三角形を描かせる命令，

正三角形　100

ならば，一辺の長さが100であるような正三角形を描かせる命令ということにしたいわけです。LOGOでは（ついでに，LISPやFORTHでは）このようなことも可能です。LOGOの場合，TO命令で命令を定義するとき，命令の名前に続けて仮引数を並べることで実現することができます。たとえば，正三角形を描かせる“正三角形”という命令の仮引数を“長さ”という名前にすることにすれば，

TO　正三角形：長さ

と入力することで，引数が付いた命令を定義することができます（：を付けるのを忘れないように)。このとき，“正三角形”という命令を実行するときに指定される数値は

：長さ

という名前で参照することができます。つまり，“正三角形”という命令の定義を

REPEAT　3 ［ FD ：長さ RT 120］

としてやれば，自由な大きさの正三角形を描かせるための命令が定義できるのです(リスト2)。ここで，少し注意があります。turbo LOGOの初期状態では，同じ名前の命令は定義できないようになっています。同じ名前の命令を定義したいときは，

MAKE　"REDEFP　"TRUE

というオマジナイを実行しておいてください。なお，先程定義したばかりの，“正三角形”という命令の実行結果を図6に示しておきましょう。

これまでの説明で，正三角形を描かせる命令を定義する方法はわかったと思います。それでは，もう一歩進んで，正多角形を描かせるための命令の定義はどうなるでしょうか。もうくどくどとは説明しませんが，そのプログラム(新しい命令の定義の仕方)はリスト3のようになります。また，このときの実行結果を図7に示しておきましょう。

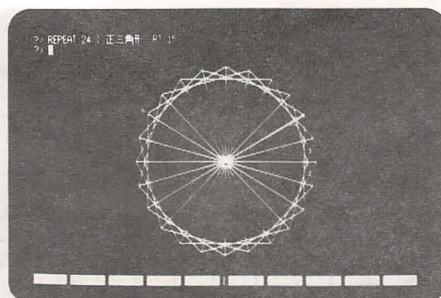



図5　正三角形を回転してみました

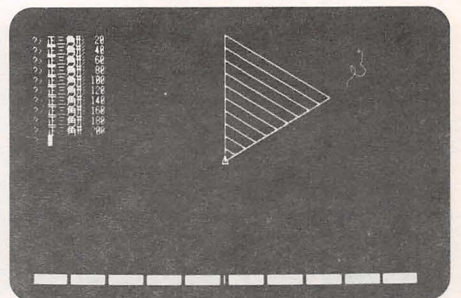



図6　正三角形，正三角形，正三角形，……

リスト3

```
?>
?>
?>
?>
?> TO 正多角形  :角の数 :辺の長さ
>> REPEAT :角の数  [ FD :辺の長さ  RT 360 / :角の数 ]
>> END
正多角形 DEFINED
?>
?>
?>
```

リスト4

```
?>
?>
?>
?>
?>
?> TO らせん  :距離 :角度
>> IF :距離 > 250 [ STOP ]
>> FD :距離  RT :角度
>> らせん  :距離 + 2  :角度
>> END
らせん DEFINED
?>
?>
?>
?>
```

4 自分自身で自分を定義

LOGOはLISPをモデルにしてあるだけあって，再帰もお手のものです。再帰についてわざわざ説明する必要はないと思いますが，念のために言っておけば，自分自身を使って自分を定義することです。たとえば，タートル君に図8のようならせん図形を描かせることを考えます。これは，

```
FD  2   RT  90
FD  4   RT  90
FD  6   RT  90
FD  8   RT  90
FD  10  RT  90
    ⋮
```

という具合に，1回前より2だけ長い距離を進んだあと90°だけ右に回転するという命令を何度も繰り返すときに得られるものです。こういう動作をする命令を定義しようとすれば，先程のREPEATを使うことが頭に浮かびますが，次のように考えることもできます。つまり，"自分自身"を定義するためには，前進し，右に回転したあと"自分自身"を再び実行すればよいことになります。これをLOGOのプログラムになおせば，

```
TO  らせん  :距離  :角度
FD  :距離    RT  :角度
らせん  :距離  +  2    :角度
END
```

となります。

まさに"らせん"という命令を定義するのに，"らせん"自身が使われているのがわかるでしょう。この"らせん"命令を使って図8の図形を描かせるためには，

```
らせん  2    90
```

を実行すればよいのです。しかし，この"らせん"命令には重大な欠陥があります。つまり，一度命令を実行させたら最後，タートル君は終わることなく図形を描き続けるのです（このときはCTRL-Cで止めるしか方法がありません)。これは"らせん"という命令の定義の中で，いつこの命令の実行をおしまいにするのかを指示してないからです。たとえば，変数"距離"の値（:距離のこと）が，250より大きくなったとき，実行を終わらせることにしましょう。このときは，プログラムの最初で，

```
IF  :距離 > 250  [ STOP ]
```

という命令を入れておきます。つまり，完璧な"らせん"というプログラムはリスト4のようになります。ここで，もう一度

```
らせん  2  90
```

を実行してみましょう。今度はちゃんと止まりましたね。

なおIFという命令はBASICのIF文と同

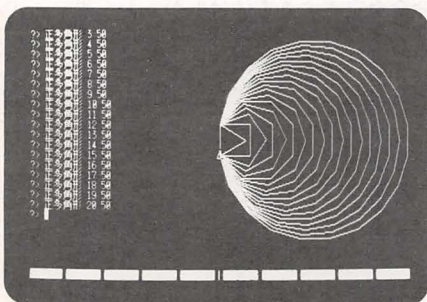



図7 いろいろな正多角形です

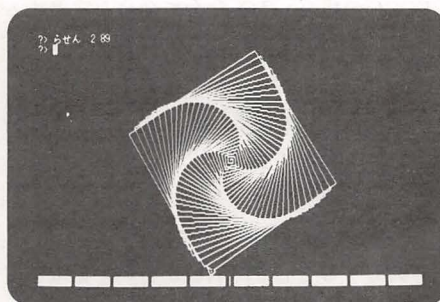



図9-A らせん 2 89

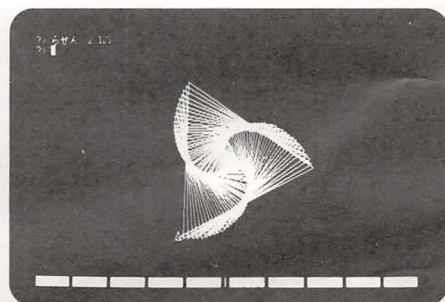



図9-B らせん 2 170

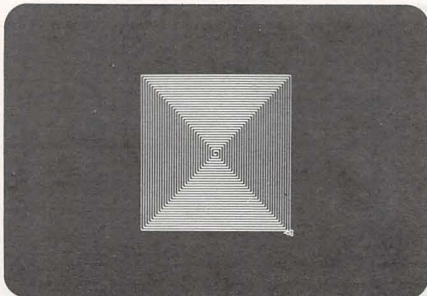



図8 らせん図形です

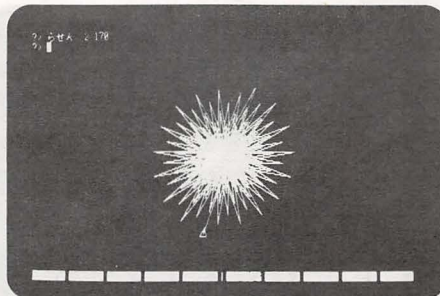



図9-C らせん 2 121

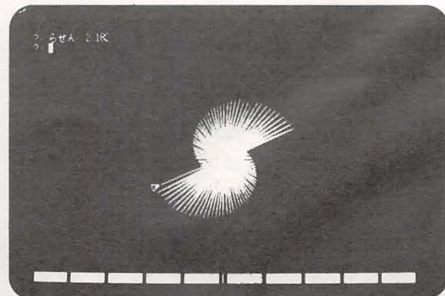



図9-D らせん 2 182

リスト5

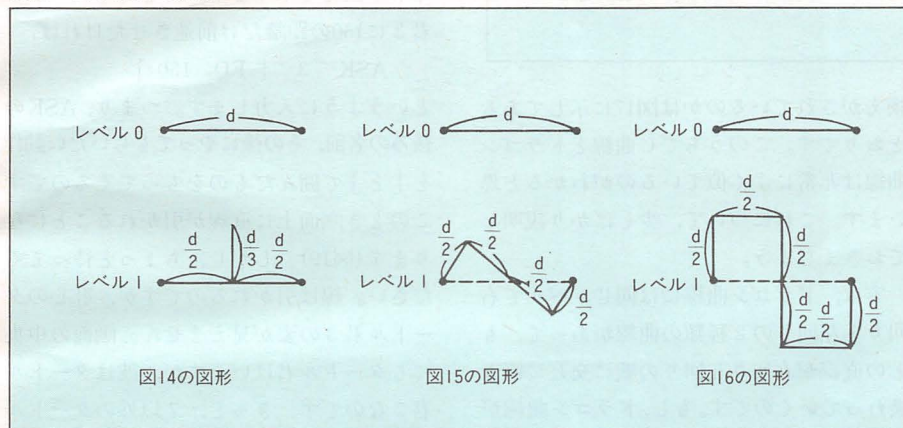
```
?>
?> PO "例 1
TO 例 1 :辺 :レベル
IF :レベル = 0 [ FD :辺 STOP ]
例 1 :辺 / 2 :レベル - 1
LT 90
例 1 :辺 / 3 :レベル - 1
RT 180
例 1 :辺 / 3 :レベル - 1
LT 90
例 1 :辺 / 2 :レベル - 1
END
?>
```

リスト6

```
?>
?> PO "例 2
TO 例 2 :辺 :レベル
IF :レベル = 0 [ FD :辺 STOP ]
LT 60
例 2 :辺 / 2 :レベル - 1
RT 120
例 2 :辺 / 2 :レベル - 1
例 2 :辺 / 2 :レベル - 1
LT 120
例 2 :辺 / 2 :レベル - 1
RT 60
END
?>
?>
?>
?>
```

リスト7

```
?>
?>
?>
?>
?>
?> PO "例 3
TO 例 3 :辺 :レベル
IF :レベル = 0 [ FD :辺 STOP ]
LT 90
例 3 :辺 / 2 :レベル - 1
RT 90
例 3 :辺 / 2 :レベル - 1
RT 90
例 3 :辺 / 2 :レベル - 1
例 3 :辺 / 2 :レベル - 1
LT 90
例 3 :辺 / 2 :レベル - 1
LT 90
例 3 :辺 / 2 :レベル - 1
RT 90
END
?>
```

図13 置き換えの原則

じで、IFの次に書かれる条件式が成立するとき、決められた命令を実行するものです。

IF ☆ [□]

という命令は☆で示される条件式が成立するとき□という命令（複数個あってもよいのです）が実行されます。また、

IF ☆ [□] [△]

という命令は、条件が成立するとき□を、成立しないとき△を実行するものです。

ところで、この“らせん”という命令はナカナカ味のある命令で、：角度という引数に渡す値をいろいろと変えてやれば、美しい図形を描かせることができます。図9にいくつかの例を載せておきましょう。このほかにも各自で試してみてください。ときには予期しないような図形が描かれることがあるかもしれませんよ。

5 1本の直線から

LOGOでは再帰的に命令を定義できることがわかったと思います。再帰とは、自分で自分を参照することでしたから、この機能を用いれば、自分の一部が自分と同じような形をしている図形（いわゆる自己相似型の図形）を簡単に描かせることができます。このような図形の一例を図10～図12に示しましょう。これらは結構複雑な図形ですが、図形を描く手順はいたって単純です。つまり、1本の直線を一定のパターンで置き換えていく操作をするだけでできあがってしまうのです。たとえば、図10を見てみましょうこれは1本の直線を

（図）

というパターンで置き換えてやるとできあがります。図10は90°回転していますが、全体や、その一部をながめてみても、どちらも

（図）

というような形をしているのがわかると思います。図10～図12の図形が、直線に対してどのような置き換えをすれば得られるのかということを、図13に示してありますが、どれもみな単純な置き換えしかやってないことがわかるでしょう。ここでは、何回置き換えが行われたのかを示すために“レベル”という言葉を用いています。つまり、もともとの直線がレベル0で、それを一定のパターンで置き換えたものがレベル1、レベル1の直線部分を前と同じパターンで

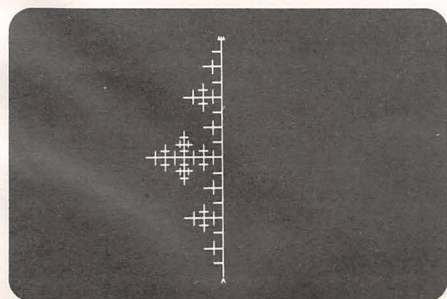



図10 自己相似型の図形（その1）

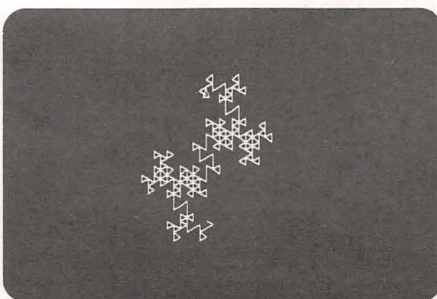



図11 自己相似型の図形（その2）

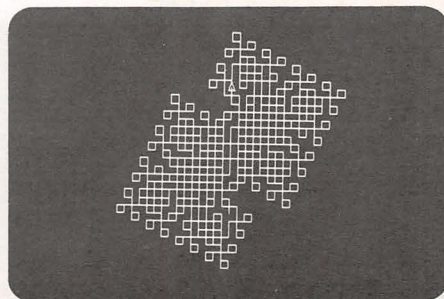
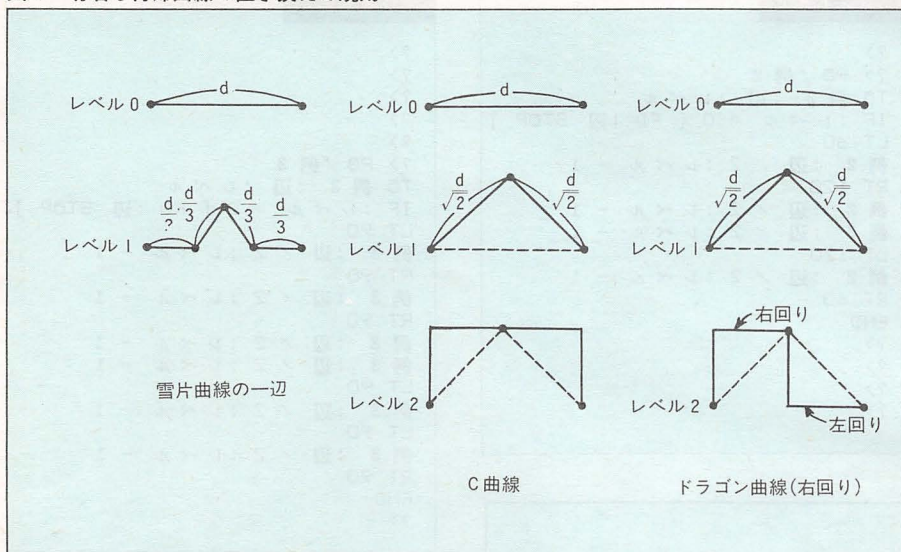



図12 自己相似型の図形（その3）

図17　有名な再帰曲線の置き換えの規則

置き換えたものがレベル2，レベル2の直線部分を前と同じパターンで置き換えたものがレベル3，……，という具合です。

これらの図形をタートル君に描かせるためには，最終的なレベルを指定して（たとえばnとしましょう），それをひとつだけレベルの小さい図形（レベルn−1）で描かせればよいのです。すなわち，レベルnの図形を描かせるためには，レベルn−1の図形を直線とみなして，パターンの図形をなぞればよいのです。ですから，図10〜図12の例を描かせるLOGOプログラムは，リスト5〜7のようになります。なお，プログラム中のLTという命令はLEFTの省略形で，タートルを左に回転させるための命令です（つまり，RIGHTの反対です）。

ところで，自己相似型の図形として有名なものに，雪片曲線（図14）の一辺，C曲線（図15），ドラゴン曲線（図16）といったものがあります。ご覧のように，これらの図形は独特な形をしていますが，これらもみな，1本の直線を特定のパターンで置き換えたものにすぎません。どのような置き換えがされているのかは図17に示してあるとおりです。このうちでC曲線とドラゴン曲線は非常によく似ているのがわかると思います。これについて，少しばかり説明しておきましょう。

実は，ドラゴン曲線には同じレベルで右回りと左回りの2種類の曲線があって，もとの直線が右回り左回りの順に交互に置き換わっていくのです。もし，ドラゴン曲線が右回りのものしかなければ，それはC曲線とまったく同じものになってしまうでしょう。原理的にはほとんど同じものなのに，その割には全然違った曲線になってしまううのですから不思議なものですね。まあ，それはともかく，リスト8に，雪片曲線，C曲線，ドラゴン曲線を描かせるプログラムを示しておきます。これからわかると思いますが，雪片曲線は“雪片曲線の辺”を一辺にして正三角形を描かせるときにできるものですね。この他にも，これまでに示してきた自己相似型の図形を一辺にして多角形を描かせてみたり，回転させたりしてみましょう。いろいろな図形ができて，なかなか楽しいものです。図18はその一例で，図11の図形を一辺にして正六角形を描かせたときにできる図形です。

6 7匹のカメを使って

turbo LOGOのタートルグラフィックスでの最大の特長は，タートルを7匹まで扱うことができることです。これについて説明しましょう。

7匹のタートル君は，名前を1，2，3，4，5，6，7と言います（そっけない名前ですね)。これらのタートル君たちを動かしたいときは，ASK命令を使って名指しで命令を伝えてやります。たとえばタートル君3に150の距離だけ前進させたければ，

ASK　3　[FD　150]

というように入力します。つまり，ASKの後ろの名前，その後にやってもらいたい動作を［と］で囲んだものをもってくるのです。このとき画面上に直線が引かれることになります(図19)。しかし，ちょっと待ってください。線は引かれるのですが，肝心のタートル君3の姿が見えません。画面の中央にもタートル君はいますが，彼はタートル君7なのです。きっと，7以外のタートル君は恥ずかしがりやなんですね。かなり前に説明したST命令で姿を見せてもらいましょう。このための命令は，もちろん，

ASK　3　[ST]

です。タートル君3は画面に紫色の姿を現しました(図20)。ここで，勘のいい人なら気付くと思いますが，7匹のタートル君たちは，自分の名前で表される色（1が青色，2が赤色，3が紫色，4が緑色，5が水色，6が黄色，7が白色というよくあるやつです）と同じ色をしているのです。これを知っていれば，画面上に何匹ものタートル君が出てきても，誰が誰だか区別することができます。

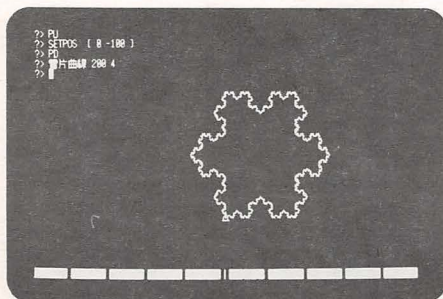



図14　雪片曲線(レベル4)

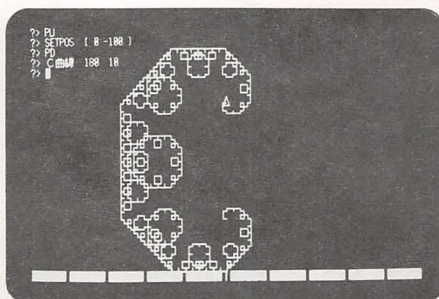



図15　C曲線(レベル10)

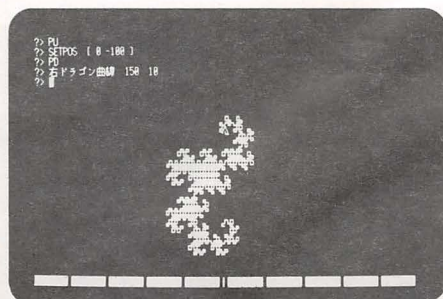
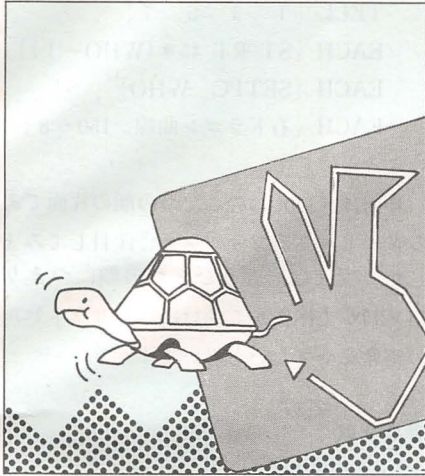



図16　ドラゴン曲線(レベル10)

このASK命令を使えば，7匹のタートル君を自由に扱うことができるのですが，それぞれのタートル君に同じ命令を与えるためには，必要なタートル君の数だけASK命令を描かなければならないのが不便です。そこで，EACH命令を紹介しましょう。

このEACH命令は，あらかじめTELL命令で指示されたタートル君に対して，同じ命令を順々に与えていきます。TELL命令というのは，必要なタートル君の名前を並べたものを［と］で囲んでやって指示します。たとえば，

TELL ［ 1 3 5 ］

を実行すれば，EACH命令ではタートル君1と，タートル君3，タートル君5に命令を与えることができます。そして，そのEACH命令というのは，それぞれのタートル君にやってもらいたい動作を［と］で囲んで指示するのです。たとえば，今の場合，

EACH ［ ST FD 150 ］

は，

ASK 1 ［ ST FD 150 ］
ASK 3 ［ ST FD 150 ］
ASK 5 ［ ST FD 150 ］

と同じことになります。ところで，TELL

リスト8

```
?>
?>
?>
?>
?> PO "雪片曲線
TO 雪片曲線 :距離 :レベル
REPEAT 3 [ 雪片曲線の辺 :距離 :レベル RT 120 ]
END
?>
?> PO "雪片曲線の辺
TO 雪片曲線の辺 :距離 :レベル
IF :レベル = 0 [ FD :距離 STOP ]
雪片曲線の辺 :距離 / 3 :レベル - 1
LT 60
雪片曲線の辺 :距離 / 3 :レベル - 1
RT 120
雪片曲線の辺 :距離 / 3 :レベル - 1
LT 60
雪片曲線の辺 :距離 / 3 :レベル - 1
END
?>
?> PO "C曲線
TO C曲線 :距離 :レベル
IF :レベル = 0 [ FD :距離 STOP ]
LT 45
C曲線 :距離 / 1.4142136 :レベル - 1
RT 90
C曲線 :距離 / 1.4142136 :レベル - 1
LT 45
END
?>
?> PO "右ドラゴン曲線
TO 右ドラゴン曲線 :距離 :レベル
IF :レベル = 0 [ FD :距離 STOP ]
LT 45
右ドラゴン曲線 :距離 / 1.4142136 :レベル - 1
RT 90
左ドラゴン曲線 :距離 / 1.4142136 :レベル - 1
LT 45
END
?>
?> PO "左ドラゴン曲線
TO 左ドラゴン曲線 :距離 :レベル
IF :レベル = 0 [ FD :距離 STOP ]
RT45
右ドラゴン曲線 :距離 / 1.4142136 :レベル - 1
LT 90
左ドラゴン曲線 :距離 / 1.4142136 :レベル - 1
RT 45
END
?>
?>
?>
```

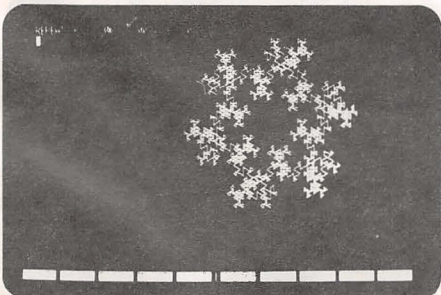



図18 図11の図形を1辺とする正六角形

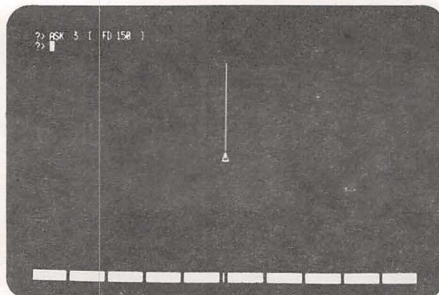



図19 タートル3の姿が見えない

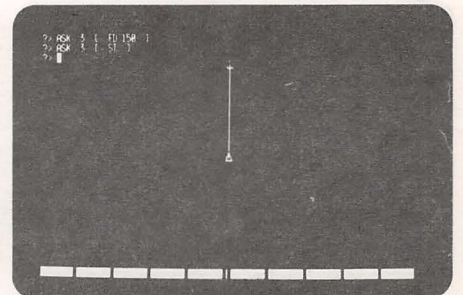
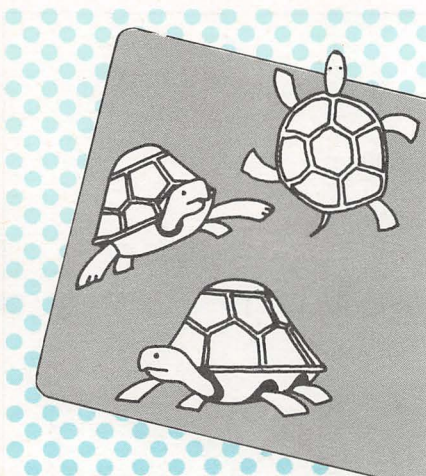



図20 タートル3が姿を現した

命令によって，どのタートル君が指定されているのかを知りたいときがあります。このために，turbo LOGOにはWHOという関数があります。この関数は，TELL命令で指定されているタートル君たちの名前のリストを値とします。

SHOW WHO

と入力してみてください。先程のTELL命令を実行した後なら，画面に

[1 3 5]

と表示されるはずです（SHOW命令は関数などの値を画面に出力するときに使う命令で，よく似た機能を持つ命令にPRINTというものもあります)。LOGOでは，このように［と］で囲まれたものをリストと呼びます。つまり，前記の例は1，3，5を要素するリストと言えます。まあ，これだけの機能しかないのならば，WHO関数の使い道はあまりないでしょう。しかし，WHO関数はもうひとつの重要な機能があるのです。

それは，WHO関数がEACH命令の［ ］の中で使われる場合です。このときは，そこのEACHで実行されるべきタートル君の名前が値になります。たとえば，先程のTELL命令に対して，

```
EACH [ FD WHO * 20 ]
```

は，

```
ASK 1 [ FD 1 * 20 ]
ASK 3 [ FD 3 * 20 ]
ASK 5 [ FD 5 * 20 ]
```

と同じことになります。これが理解できれば，

```
TELL [1 2 3 4 5 6 7]
EACH [ ST ]
EACH [RTWHO*(360/7) FD150]
```

で何が起きるのかわかりますね。もちろん図21のような図形になります。さて，今までは，タートル君の動いた跡は白色の軌跡が残りましたが，これはSETPC命令で変更することができます。つまり，

SETPC ☆

という形で使用し，☆のところには色を表す数値（これはタートルの色と名前の関係と同じ）を書きます。つまり，

```
EACH [ SETPC WHO ]
```

という命令を実行すると，タートル君の色と，その軌跡を同じ色にすることができるのです。せっかくですから，それぞれのタートル君たちが，自分自身の色の軌跡を残しながら図形を描いていく例を挙げておきましょう。最初は，

```
TELL [1 2 3 4 5 6 7]
EACH [SETPC WHO ST]
REPEAT 1000[EACH[RT
          RANDOM 360 FD 10]]
```

です。これは，乱数関数RANDOMを使ってタートル君の向きを決めて，ひたすら前進させるだけの命令です。タートル君たちはみんな勝手に動きますから，REPEAT命令で1000回も繰り返すと，画面の上で7つの色が混じりあってなかなかきれいです(図22)。もう少しまともな例としては，ドラゴン曲線を4匹のタートル君に描かせます(図23)。これは，

```
TELL [1 3 5 7]
EACH [ST RT 45*(WHO-1)]
EACH [SETPC WHO]
EACH [右ドラゴン曲線 150 8]
```

によるものです。

今回は，LOGOのふたつの顔の片面であるタートルグラフィックスに注目してみました。次回は，もうひとつの顔，つまりLOGOがLISPから受け継いだリスト処理の紹介をやりたいと思います。

参考文献

1) 『turbo LOGO(漢字版) ユーザーズマニュアル』
2) 小谷善行：『パソコンLOGO プログラミング』，東海大学出版会，1984年
3) ピーター・ロス（倉谷直臣 訳)：『やさしいLOGO入門』，啓学出版，1985年
4) マーチン・レッサー（奥和宏 訳)：『パソコンのためのロゴ』，啓学出版，1985年
5) こうもとやすひこ：『MZ-LOGO 入門タートルグラフィック』，Oh！MZ 1983年9月号
6) こうもとやすひこ：『LOGO とグラフィックス』，Oh！MZ 1984年 2月号

1)はもちろんturbo LOGOのマニュアル。基本的には命令の解説だけしか載っていないが，プリミティブ（基本命令）のフルネームとその読み方（なんとカタカナ）が付録となっているのが目新しい。このマニュアルだけではLOGOの概要を知るのは難しいが，2)～4)あたりを参考にすればよいだろう。2)はFM-LOGO，3)はTerrapin LOGO，4)はLCSI LOGOをもとにして書かれているが，LOGOの文法はどれも似たようなもの（Terrapin LOGOは他に比べて少し異なるが）なので混乱はしないであろう。とくに4)はリスト処理について詳しく述べられている。5)，6)はOh！MZで取り上げられたLOGOのタートルグラフィックスに関する記事。MZ-LOGOは turbo LOGOとロウアーコンパチ（こんな言葉はあるのかな）であるから，これらの記事のサンプルは（プリミティブの名前を小文字から大文字に直すだけで）ほとんどすべてturbo LOGOで動かすことができる。

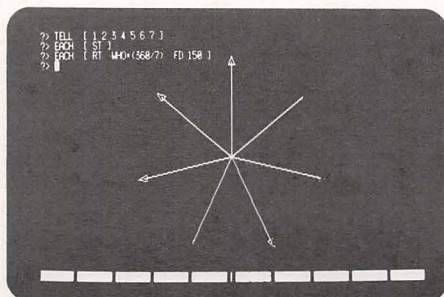



図21 7匹のカメです

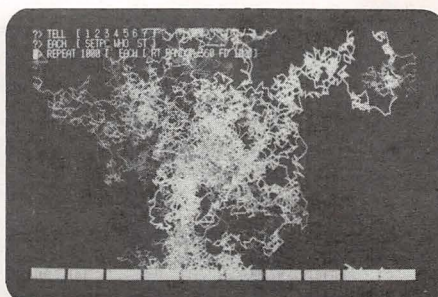



図22 足の向くまま気の向くままのグチャグチャ曲線

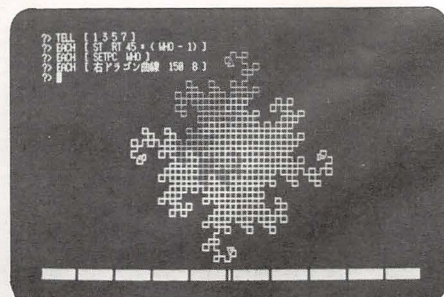



図23 4つのドラゴル曲線を組み合わせました

BASIC"行動学"入門

皆さんはBASICをどのように使っていますか？　ゲームや実務プログラムなど目的を達成するための手段でしょうか。マシン語やほかの言語を勉強するためのひとつの過程でしょうか。それともBASICそれ自体が遊びの道具になっているのでしょうか。

人によって考え方はさまざまでしょうが，それでもBASICをまったく使わないというパソコンユーザーはほとんどいないと思います。アクション，アドベンチャー，ロールプレイなんでもこいのゲーマーであっても，アセンブラがビンビンのマシン語プログラマであっても，LispやProlog，Forth，Cを使いこなす言語博士であっても，ハンダごてバンバンのハードマニアであっても，やっぱりBASICは使うのです。

パソコンに標準装備の言語だから，という理由だけではありません。BASICは遅い，メモリを食うという大きな欠点を持ってはいますが，Beginer's All-purpose～の名のとおり誰もが多方面に利用できるツールであり，そのマシンの特長をもっともよく表したソフトウェアなのです。また，パソコン雑誌に掲載されるプログラムの多くはBASICで書かれています。機種によって方言があるといっても基本的なところに違いはないので，ちょっと頑張ればほとんどすべてのプログラムが利用できます。

BASICは私たちにとってもっとも手軽なマシンとのインタフェイスです。使い方しだいでパソコンライフが大きく変わります。今回の特集ではBASICの基本を徹底して追求し，多角的なアプローチでプログラミングを楽しむためのヒントがいっぱいです。これを参考に新たなる"行動"を起こしてみませんか。

第1部　私のBASIC行動学

第2部　BASIC活用講座

付録　他誌を10倍楽しむ方法

暇プロの楽しみ

Izumi Daisuke
泉 大介

つれづれなるままに，日ぐらしパソコンにむかひて，心にうつりゆくよしなし事を，そこはかとなく打ちつくれば，あやしうこそものぐるほしけれ。

人は時に暇をもて余してしまうことがあります。何をしようという訳でもなく，しなければいけないこともない。何とはなしにコンピュータの電源を入れて，BASICを起動してみる。さて，何をしようか。

暇プロ（暇々プログラミング）はこうしてスタートします。いくら暇を持て余していても，時間が無尽蔵にあるはずもないので，一本の完結するプログラムを作るのはまず無理なこと。もっともゲームを作ろう！などと思い立ってコンピュータに向かうのは，暇プログラマの資格なしですな。

リスト1 蛇の迷走

```
1000 DIM SX(10),SY(10)
1010 SX(10)=1 : SY(10)=0 : CONSOLE C40 : PRINT CHR$(12)
1030 REM -----------
1040 LE=RND(1)*11 : VX=INT(RND(1)*3)-1 : VY=INT(RND(1)*3)-1
1050 FOR I=1 TO LE
1055 IF (VX=0)*(VY=0) THEN VX=1
1060 IF SX(10)+VX>39 THEN VX=-1
1070 IF SX(10)+VX<0 THEN VX=1
1080 IF SY(10)+VY>23 THEN VY=-1
1090 IF SY(10)+VY<0 THEN VY=1
1100 FOR J=0 TO 9
1110 SX(J)=SX(J+1) : SY(J)=SY(J+1)
1120 NEXT
1130 SX(10)=SX(10)+VX : SY(10)=SY(10)+VY
1140 GOSUB 5000
1150 NEXT
1160 GOTO 1030
1170 REM -----------
4990 END
5000 CURSOR SX(0),SY(0):PRINT " ";
5010 FOR J=1 TO 9
5020 CURSOR SX(J),SY(J):PRINT "o";
5030 NEXT
5040 CURSOR SX(10),SY(10):PRINT "@";
5050 RETURN
```

短時間で簡単に

それでは，暇プログラマはコンピュータに向かって何をするのか。これは例を見てもらったほうがわかりやすいでしょう。

リスト1。これは画面の中を所狭しと蛇が走りまわるプログラムです。ただ走りまわるだけ。

このとき私の頭の中には，迷路の中を蛇が走りまわって何かするゲームを作ろうかなァという思惑があったのです。けどそんなものを作るには，ちゃんと紙の上で練ってからでないと，後々泣きを見るのは明らかでしょう。とりあえず蛇だけでも動かしてみるか。これが暇プロの極意ですな。コーヒーでも入れて，さぁーて，ともものの10分ほどで蛇は動き始めます。

暇プロの魅力はここにもあって，即動かして楽しむことができるのは，長いプログラミングにはない良さです。3分も見ていると飽きてきて，障害物などを中に入れてやりたくなります。ついでにブツかったときの衝突音も欲しくなります。そうこうするうちに夜は更けてきて，「もー寝よーっと」。私はこのとき，習慣的にプログラムをNEWしてしまいます。戯れは戯れであって，あとからそれを改良していこうという気はサラサラないからです。

初心者と呼ばれる方々にとっては，ずいぶんもったいないことをするなァと思われるかも知れません。けれど，今日ここまで作る段階で得たノウ・ハウは，しっかり自分のものになっているでしょう？ それが大切なのですよ。こうして得たテクニックは，次に何かを作ろうとしたときに必ず役に立つでしょう。

暇プロの題材はどこにでも転がっているものです。リスト2は，某テレビ漫画を見ていたときに思い立ちました。

登場人物が何かにブツかり気絶すると，必ずといっていいほど頭のまわりを星か何かがくるくると飛びまわりますね。そのシーンを見ているうちに，この程度のことなら……と作ってしまったのです。

座標がデータ文で蓄えてありますが，これを見て，「どこが暇プロだ！ 画倒くさいじゃないか」とおっしゃる方があるかも知れませんね。甘い。誰もご丁寧に三角関数など使って計算したりしません。そーゆー面倒な方法は暇プロの精神（私の中にあるだけです。念のため）に反する。

ではどうやるかというと，まず画面のドマンナカ(19,11くらいが適当でしょう)にデンと♥を書きます。そのまわりに，いかにもそれらしく♠をまぁるく配置。あとはカーソルを動かしながら，イチ，ニイ……と座標を数えるだけです。得た座標をDATA文中に書き込んで，ハイおしまい。簡単でしょう？

暇にまかせてコンピュータに向かい，プログラミングをする。そのときに「うっ，面倒くさい」と思ったら最後，それ以上続け

る気がしなくなってしまう私にとって，こういうてきと一にその場をやりすごす手法というのは重宝するのです。

このプログラムに少し手を加えれば，太陽系もできるでしょう。もっとも惑星を9個も回すのでは，速度の問題がチョットという気もしますが，また，隅のほうに○を書いて，さらに長い楕円軌道でピリオドを飛ばしてやれば「ハレー彗星地球に接近！」なんてことも可能でしょう。

ひとつの手法を応用するだけで，このように暇プロの世界はさらに広がっていくのです。

ちょっとしたヒントに改良を加えて

リスト3のプログラムが生まれたいきさつはこうです。私の友達のひとりが，めでたく誕生日を迎えたときに，これはひとつ，コンピュータでHAPPY BIRTHDAYをやってやらねばなるまい，と考えたのが事の起こりでした。そのままパッと表示してしまうのは，いとわろしでしょう。1行ずつスクロールアップするのは，誰でも考えつきそうでおもしろくない。かといって横スクロールは，マシン語でも使わない限り遅すぎて使いものになりません。う～む，困った困った。

そこでふと思い出したのが，以前計算機実習（大学に行けばわかります。受験生の皆さん頑張ってください）のときに目にしたデモです。そのままでは処理が面倒なところがあったので，さらに簡素にして実現してみたというわけです。

左から，あるいは右から，文字が1行ずつ横スクロールしてきて，画面を作りあげていきます。この手法は以来気に入ってしまい，ことあるごとにRUNしては飽きもせずに眺めています。

そのままでは寂しかったので，リスト3ではBEEP音を入れてあります。どうですか。いとをかし，でしょ？

リスト2　星が飛びまわる

```
100 CONSOLE C40
110 CURSOR 17,10 : PRINT "♥"
120 CURSOR 5,15 : PRINT "Daisuke, flying around Yukie"
130 X=14 : Y=8
140 FOR I=1 TO 10
150 XB=X : YB=Y
160 READ X,Y
170 CURSOR X,Y : PRINT "♠";
180 CURSOR XB,YB : PRINT " ";
190 FOR J=0 TO 30 : NEXT
200 NEXT
210 RESTORE
220 GOTO 140
1000 DATA 16,7,18,7,20,8,21,10,20,12
1010 DATA 18,13,16,13,14,12,13,10,14,8
```

リスト3　HAPPY BIRTHDAY

```
1000 CONSOLE C40
1010 OFST=0 : GOSUB 1100
1020 OFST=6 : GOSUB 1100
1030 OFST=12 : GOSUB 1100
1040 USR($0F14)
1050 END
1100 REM ==========
1110 FOR I=0 TO 4
1120 READ A$ : USR($0F14)
1130 IF I-INT(I/2)*2 THEN 1200
1140 REM ----------
1150 FOR J=1 TO 39
1160 CURSOR 0,OFST+I
1170 PRINT RIGHT$(A$,J);
1180 NEXT
1190 GOTO 1290
1200 REM ----------
1210 FOR J=1 TO 39
1260 CURSOR 39-J,OFST+I
1270 PRINT LEFT$(A$,J);
1280 NEXT
1290 NEXT
1300 RETURN
1990 REM ==========
2000 DATA "
2010 DATA "
2020 DATA "
2030 DATA "
2040 DATA "
2050 DATA "
2060 DATA "
2070 DATA "
2080 DATA "
2090 DATA "
2100 DATA "
2110 DATA "
2120 DATA "
2130 DATA "
2140 DATA "
```

すぐに作ってすぐ遊べ

私は実は非現代人であります。というのは，情報化社会から取り残されているからです。つまり，私はテレビを持っていないのです。ところが隣の住人は文化人で，流行に遅れもせずテニスをやっているらしい。で，夜中になると，テニスのゲーム中継をテレビ観戦し始めるわけですね。悲しいかなわがおんぼろアパートでは，その音がよく聞こえるのです。さらに追い打ちをかけるかのごとく，MZ-2000ではテレビが見れない！「X1は見れるじゃねーか。てめーこの野郎一根性出せ！ 男は気合いだ！」とおたけっても，無理なものは無理なので，モンモンとした気分を抱えつつ，私の指はテニスゲームを作りました。

ひとこと申し上げます。私は独身です。

品行方正，勤勉な人間です。というわけで，同棲もしていません。

つまり，テニスゲームを作っても２人では遊べないわけです。とあたかもそれが理由であるかのごとく，リスト４はひとりテニス。スカッシュをひとりでやるようなものとなっています。

本当は，すぐに作って，すぐさま遊びたかったからで，蛇のときでも書いたように私の暇プロにとってこれは命なのです。

このテニスゲームも，私の暇プロの特徴をよく表しています。すなわち，シンプルイズベスト。余計なものは一切付いていません。これを改造して，相手をつけてみたり，得点を表示するようにすれば，ゲームがひとつ出来あがりとなります。

10分も遊んでいれば飽きてきて，改良に手を出し始めるところなのですが，皆さんが暇プロで改造してもらえればと思い，あえて最初に作ったものを載せてみました。自分だけのテニスゲーム。挑戦してみてください。

気楽なお付き合いを

チョコチョコっと作ってRUN。ピッ！ ERROR。あれ〜ェ，おかしいな。何がマズイんだ？ あっそーかそーか。手直しをして再びRUN。う〜ん，どうも気にくわん。ここをチョッと変えてと。とまあこのようにして私の暇プロは続いていきます。決して肩に力が入ったり，意気込んでやりはしません。

余暇を音楽を聴いて過ごす人。絵を見て過ごす人。バイクを転がしに出かける人。それと同じ調子で，私は短いプログラムを作っては壊し，壊しては作り，手直ししてRUN。まァこんなもんだろ。そして飽きたらNEW。それこそ，つれづれなるままに，日ぐらし，です。

プログラムを作って，それで飯を食っているわけではありません。趣味は趣味。盆栽をいじって目を細めている人のように，コンピュータと付き合っていきたい。これが私の主張です。

さて，私の部屋には目覚まし時計があります。卒業記念に贈られたもので，デザインも気に入っているのですが，１年ほど前からストライキを起こしていてベルが鳴らないという困りものです。連日夜が遅いため，朝は目覚ましなしでは起きることは不可能。

リスト４ ひとりテニス

```
100 PRINT CHR$(6) : REM X1=12
110 GOSUB 1000
120 REM ==========
130 FOR L=0 TO 1
140 GET A$ : IF (A$="4")+(A$="1") THEN 160
150 GOTO 190
160 ON 1-(A$="1") GOSUB 1200,1300
170 CURSOR LX,LB : PRINT " ";
180 CURSOR LX,LY : PRINT "┣";
190 NEXT
200 REM ----------
210 XB=BX : YB=BY
220 IF BX+VX<1 THEN VX=1
230 IF BY+VY<1 THEN VY=1
240 IF BY+VY>23 THEN VY=-1
250 BX=BX+VX : BY=BY+VY
260 IF BX<>38 THEN 400
270 IF (BY>LY-3)*(BY<LY+3) THEN 300
280 MUSIC "-E4-D-C" : GOTO 100
290 REM ----------
300 MUSIC "A0"
310 VX=-1
320 REM ----------
400 CURSOR XB,YB : PRINT " ";
410 CURSOR BX,BY : PRINT "●";
420 GOTO 130
430 END
1000 REM ==========
1010 FOR I=1 TO 38
1020 CURSOR I,0 : PRINT "—";
1030 CURSOR I,24 : PRINT "—";
1040 NEXT
1050 FOR I=1 TO 23
1060 CURSOR 0,I : PRINT "|";
1070 NEXT
1080 CURSOR 0,0 : PRINT "┌";
1090 CURSOR 0,24 : PRINT "└";
1100 REM ----------
1110 LX=39 : LY=INT(RND(1)*23)+1
1120 BX=38 : BY=12 : VX=-1 : VY=INT(RND(1)*2)*2-1
1130 DEF KEY(10)=POKE $952,1667
1140 POKE $952,0
1150 CURSOR LX,LY : PRINT "┣"
1160 TEMPO 7
1170 RETURN
1180 REM ==========
1200 LB=LY : LY=LY-1
1210 IF LY<1 THEN LY=1
1220 RETURN
1290 REM ----------
1300 LB=LY : LY=LY+1
1310 IF LY>23 THEN LY=23
1320 RETURN
1330 REM ==========
```

そこで，「明日は目が覚めるまで寝ていよう」と心に決めた日を除いて，毎晩私が打ち込むのがリスト５のプログラムです。毎晩打ち込むわけですから，早く打ち込めるようにと，ムダな飾りはまったく付けてありません。

さァ，寝るか。と思い立つと，まずBASICを起動します。MZ-1Z001はテープ版ですので，ロードの時間を利用して歯磨きを済ませてしまいます。そしてチョコチョコッと打ち込んで，RUN。現在時刻を打ち込み，起きたい時刻を入力。画面はクリアされるので，焼けてしまうといった心配はないでしょう。翌朝，味もそっけもない音で，

リスト５ 目覚まし

```
10 INPUT TI$
20 INPUT TM$
30 PRINT CHR$(6):TEMPO 7
40 IF TI$<>TM$ THEN 40
50 MUSIC"A1":GOTO 50
```

見事に目を覚ましてくれます。

ただひとつ問題なのは，停電するとすべてがパアになってしまうことですが，これは仕方がないでしょう。

夜も更けてきました。風邪など引かないよう，暖かくしてコンピュータとお付き合いください。

おやすみなさい。

グラフィックで遊びたい

Horiuchi Yasuhide
堀内 保秀

パソコンによるグラフィックと聞いて，皆さんは何を連想するでしょうか。お絵描き？ 高速3Dグラフィック？ みんなツールが必要ですし大変そうですね。もっと簡単にできるグラフィックの遊びはないんでしょうか。あります，あります。皆さんもやったことがあるでしょう。画面にランダムに点や線を描いたりしたことが。これにちょっと計算を加えるだけで，おもしろい模様が描けるのをご存じですか。こんな遊びこそ，手軽にできるもっともパソコンらしいグラフィックの楽しみではないでしょうか。

LINEで遊ぶ

まずはリスト1-Aのプログラムを走らせてみてください。リストはHuBASIC用ですので，ほかのBASICをお使いの方は各自変更してくださいね。320ドットモードでやる場合は，XMの値を319にすればOKです。

ただ単にLINEを引いているだけなのに模様ができたでしょう。パソコンのLINEが完全な直線ではなく階段状になっているために，このような模様が描けるんですね。STEPの値を変えることにより少し感じの違った模様になるでしょう。もし，手元に別の機種があったり，友達が別の機種を持っていたら，同じプログラムを走らせてみてください。機種において違った模様になることがあります。色などを変えていろいろ試してみてください。

リスト1-Bは1-Aをちょっとだけ変更したものですが，ここではLINEを引くときにXORを使っています。XORのない人はゴメンしてください。ここでXORについて簡単に説明をしておきましょう。XORはeXclusive OR（排他的論理和）のことで，この場合，引こうとする直線上で，すでに点の打ってあるところはその点を消し，点のないところへは点を打つという動作をします（厳密に言えばこれを各プレーンごとに行います）。走らせてみると不思議な模様を描きますね。どうしてこのようになるのでしょう。先ほどXORを指定すると点のあったところの点を消していくと書きましたが，これはすなわちLINEが偶数回重なった部分を消していくということですから，まったく重ならない部分はそのまま残り，2つ重なった部分は消え，3つ重なった部分は残り……というわけで，このような不思議な模様になるんですね。ループにSTEPを入れたりあるいは色を変えたりして楽しんでみてください。

SIN関数を使って

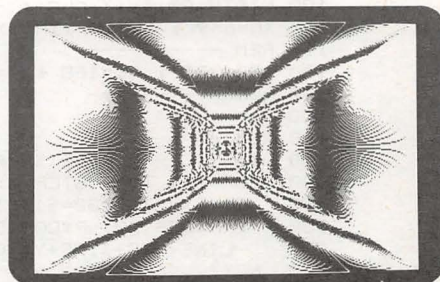
次にリスト2のプログラムを見てください。これはサイン波にノイズ波を入れて歪波を合成するものです。ここでノイズ波は基本となるサイン波に対する位相（角度のずれ）と周波数の倍率と大きさの3要素で決定します。途中LINEで同じ座標を結んでいるところがありますが，これはFOR～NEXTループ中のLINE文での始点を設定するためのものです。このようにLINE文を使うと，PSETを使ってグラフを描いたときのように細切れになることなく素早く曲線が描けるのです。

次はリスト3のプログラムです。さっそく走らせてみましょう。描けた模様に見覚えのある人も多いと思います。そう，スピログラフです。内側に向かって歯のついた丸い枠の中にそれより小さな歯車を置いて，歯車にあいている小さな穴にペンを入れてグルグル回して模様を描くというあれですよ。リスト3はそのオモチャをシミュレートしたものです。R0は枠の半径，R1は歯車の半径，Pは歯車の中心からペンまでの距離で，現実的に考えるとこの3つの値にはR0＞R1＞Pの関係が成り立っていなければなりません。しかし，何もこの関係を

リスト1-A LINEで遊ぶ（1）

```
100 WIDTH 80:INIT:CLS 4
110 XM=639
120   FOR X=0 TO XM STEP 4
130    LINE(X,0)-(XM-X,199)
140   NEXT
150   FOR Y=0 TO 199 STEP 2
160    LINE(0,199-Y)-(XM,Y)
170   NEXT
180 END
```

リスト1-B LINEで遊ぶ（2）

```
100 WIDTH 80:INIT:CLS 4
110 XM=639
120   FOR X=0 TO XM
130    LINE(X,0)-(XM-X,199),XOR
140   NEXT
150   FOR Y=0 TO 199
160    LINE(0,199-Y)-(XM,Y),XOR
170   NEXT
180 END
```

LINE文で描かれた模様

満たしていなければならないということはありません。むしろ，この関係を満たさない数値のほうがおもしろいかもしれません。そんな非現実的なことを簡単に再現してくれるのがコンピュータのいいところであるのですから。なお，このプログラムでは各変数の値をいっさいチェックしていませんので，ただ楕円が描けるだけのときもあります。めげずにいろいろ試してください。また，終了のチェックもしていないので，描き終えたと思ったらBREAKしてください。本来チェックできることなのですが，プログラムを簡潔にするためにあえて省きました。必要なら皆さんで追加してください。

リスト2とリスト3では主にSIN関数しか使っていませんが，かなりいろいろな模様が描けることがおわかりいただけたでしょう。ほとんどすべての関数はSIN関数の和として表すことが可能です（フーリエの定理)。まだSIN関数を習っていない読者の方も多いと思いますが，高校へ行って習うことになったら尻込みせずに理解するようにしましょうね（実は私もこれにかなり苦しめられたのです)。

物体を表示する

さて次は物理のシミュレーションです。「数学の次は物理か……」といってため息をつくことはありません。計算するのはあなたではなく目の前のコンピュータなのですから。リスト4のプログラムはある高さから投げ出されたボールが床や壁に衝突して跳ね返される様子を再現します。Gは重力加速度で，地球上では約9.8，月面では地球上の約6分の1の値になる物理定数。Eは跳ね返り係数と呼ばれるもので0≦E≦1の値をとります。V0は初速度，Y0は投げ出される高さ，Dは投げ出す角度で－90°≦D≦90°の値をとり，値が正のときは上向き，負のときは下向きに投げ出されることを意味します。SPは何秒おきに状態を画面に表示するかを表しますが，衝突の瞬間を表示するために必ずしもSP秒ごとのものが描かれるとは限りません。ATを0以外の数にするとボールの軌跡が表示され，ストロボ写真のようになります。

ちょっと難しかったかもしれませんが，変数の値をいろいろ変えてやってみてください。

リスト5のプログラムは立体形を表示するものです。とはいっても立体テレビなるものがあるわけではないので，当然3D→2D変換をして平面上に表すことになります。まず座標の変換ですが，3次元上での座標を（X, Y, Z)，2次元上での座標を（U, V）として次のようにします。

$$U=X-Y$$

$$V=X+Y/Z+Z$$

本来もう少し複雑なのですが簡略化のためにこうしておきます。次に隠線処理（裏に隠れて見えなくなるべき線を消すこと）です。このプログラムでは図形を前面から後面に向かって表示しているので，各縦線上で新しく計算したVの値がそれまでに表示していた点のVの値の最大値と最小値の間にある場合は，その新しい点が後ろ側の点であるとして，それを表示しないようにしてあります。ちょっと難しかったかな？これを実現するため，T(U)にそれまでの最大値，B(U)に最小値をそれぞれ記録しています。

表示する関数はDEF FNZ(X, Y）で定義しています。ここを書き換えることによってさまざまな図形が描けるはずです。要はXとYの式になっていればいいのですが，画面からはみ出してしまうことも考えられるので注意してください。

なお，プログラム中に出てくる変数SPは計算するステップ値で，この値を大きくすると実行速度が速くなります。しかし，大きくしすぎると絵が雑になり，何を描い

リスト2 サイン波合成

```
100 WIDTH 80:INIT:CLS 4
110 XM=639
120 SP=2
130 REM ----------
140 PH=30 :REM PH=ｲｿｳ
150 FR=3  :REM FR=ﾊﾞｲﾘﾂ
160 LV=50 :REM LV=ｵｵｷｻ (% ﾃﾞ ｱﾗﾜｽ)
170 REM ----------
180 Y0=99
190 YG=60
200 LINE(0,Y0)-(XM,Y0)
210 REM ----------
220 DEF FNY1(X)=SIN(X*π/180)*YG
230 DEF FNY2(X)=SIN(FR*(X+PH)*π/180)*YG*LV/100
240 REM ----------
250 Y=Y0-FNY1(0):LINE(0,Y)-(0,Y),PSET,5
260   FOR X=0 TO XM STEP SP
270     Y=FNY1(X)
280     LINE-(X,Y0-Y),PSET,5
290   NEXT
300 REM ----------
310 Y=Y0-FNY2(0):LINE(0,Y)-(0,Y),PSET,2
320   FOR X=0 TO XM STEP SP
330     Y=FNY2(X)
340     LINE-(X,Y0-Y),PSET,2
350   NEXT
360 REM ----------
370 Y=Y0-(FNY1(0)+FNY2(0)):LINE(0,Y)-(0,Y),PSET,6
380   FOR X=0 TO XM STEP SP
390     LINE-(X,Y0-(FNY1(X)+FNY2(X))),PSET,6
400   NEXT
410 END
```

リスト3 スピログラフ

```
100 WIDTH 80:INIT:CLS 4
110 XM=639 : XC=INT(XM/2)
120 REM ----------
130  R0=200 : R1=160 : P=125 : R=R0-R1 : YG=.5
140 REM ----------
150       SP=4
160 REM ----------
170 LINE(XC+R-P,99)-(XC+R-P,99),PSET,0
180   FOR I=0 TO 1 STEP 0:D=D+SP
190    S1=D*π/180:S0=S1*R1/R0
200    X=R*COS(S0)-P*COS(S1):Y=R*SIN(S0)+P*SIN(S1)
210     LINE-(XC+X,99+Y*YG)
220   NEXT
230 END
```

リスト4 弾むボール

```
100 WIDTH 80:INIT:CLS 4
110 XM=639 : XR=XM-10
120 REM ----------
130   G=9.8 : E=.7
140 REM ----------
150    SP=.2 : AT=0
160 REM ----------
170   V0=60 : Y0=100 : D=45
180 REM ----------
190 DEF FNT(V,Y)=(V+SQR(V*V+2*G*Y))/G:DEF FNY(V,Y,T)=V*T-.5*G*T*T+Y
200 DEF FNT1(V,Y)=(V-SQR(V*V+2*G*Y))/G
210 S=D*π/180:VY0=V0*SIN(S):VX0=V0*COS(S)
220 VY=VY0:VX=VX0:X0=10
230 T=-SP
240 REM ----------
250  FOR I=0 TO 1 STEP 0
260   T=T+SP
270   X=VX*T+X0:Y=VY*T-.5*G*T*T+Y0
280 REM ########## ｼｮｳﾄﾂ ｼｮﾘ ###########
290 REM ---- ﾕｶ ----
300   IF Y>=10 THEN 340
310   T=FNT(VY,Y0):X=VX*T+X0:Y=10:C=(C+1) MOD 7
320   VY=-(VY-G*T)*E:Y0=10:X0=X:T=0:GOTO 460
330 REM ---- ｶﾍﾞ(ﾐｷﾞ) ----
340   IF X=<XR THEN 380
350   T=T-(X-XM)/VX:X=XR:Y=FNY(VY,Y0,T):C=(C+1) MOD 7
360   VX=-VX*E:X0=XR:VY=VY-G*T:T=0:Y0=Y:GOTO 460
370 REM ---- ｶﾍﾞ(ﾋﾀﾞﾘ) ----
380   IF X>=10 THEN 420
390   T=T+X/VX::X=10:Y=FNY(VY,Y0,T):C=(C+1) MOD 7
400   VX=-VX*E:X0=10:VY=VY-G*T:T=0:Y0=Y:GOTO 460
410 REM ---- ﾃﾝｼﾞｮｳ ----
420   IF Y=<189 THEN 460
430   T=FNT1(VY,Y0-189):X=VX*T+X0:Y=189:C=(C+1) MOD 7
440   VY=-(VY-G*T)*E:Y0=189:X0=X:T=0:GOTO 460
450 REM ----------
460   CIRCLE(X,199-Y),5,C+1
470   IF AT=0 THEN CLS 0
480  NEXT
490 END
```

リスト6 真円チェック

```
100 WIDTH 80:INIT:CLS 4
110 XM=639
120 YG=.5
130  FOR D=0 TO 360 STEP 5
135   S=D*π/180
140   X=COS(S)*50:Y=SIN(S)*50
150   PSET(X+XM/2,Y*YG+99)
160  NEXT
170 END
```

ているのかわからなくなりますので適当に調節しましょう。また，YGは画面の縦と横の比が1対1になっていないことからくる歪みを補正するためのものです。リスト6のプログラムを走らせて真円が表示されるように，YGの値を設定してください。

リスト5 3次元グラフ

```
100 WIDTH 80:INIT:CLS 4
110  XM=639
120 DIM T(XM),B(XM)
130  FOR I=0 TO XM
140   T(I)=-1:B(I)=199:LOCATE 0,0:PRINT XM-I
150  NEXT
160 REM ----------
170 CLS
180 DEF FNZ(X,Y)=30*COS(π*X/20)*SIN(π*Y/20)
190 SP=1
200 REM ----------
210  FOR Y=0 TO 80 STEP SP
220   FOR X=0 TO XM/4 STEP SP
230    Z=FNZ(X,Y)
240    P=X-Y
250    U=XM/319*P+XM/2
260    Q=Z+X+Y/2
270    V=190-Q
280     IF U<0 OR U>XM THEN 370
290     IF V<0 OR V>199 THEN 370
300     IF V>B(U) AND V<T(U) THEN 370
310     IF V<B(U) THEN B(U)=V
320     IF V>T(U) THEN T(U)=V
330     C=6
340     IF Z<0 THEN C=5
350     IF Z>0 THEN C=7
360    PSET(U,V,C)
370   NEXT
380  NEXT
390 END
```

以上BASICで簡単にできるグラフィックのプログラムをいくつか紹介しました。これほど短いものでもかなりのことができるのがわかっていただけたと思います。しかし，これはまだほんの入口にすぎません。パソコングラフィックの世界はもっと奥が深いのです。お絵描きばかりがグラフィックじゃない！ 皆さんのアイデアと工夫で大いに“シンプルグラフィック”を楽しんでください。

BASIC探検隊

Nakagawa Norichika
中川 智哉

正しいベンチマークテスト

パソコン雑誌ではリスト1-Aのようなベンチマークテストをよく見かけます。このようなスピードテストの目的は何でしょう。

もっともよくあるパターンは“マシン/BASICの比較”であり読者の多くもこれを望んでいるようです。しかし，ベンチマークテストの結果はマシンの性能のすべてではありません。「そんなことわかっている」という人もたくさんいるでしょうが，BASICの，それも一部のコマンドのスピードだけを気にする人がいるのもまた事実です。確かにマシンの性能はBASICのスピードに十分反映されますが，同じ性能であれば，メモリを使えば使うほど，命令数を少なくするほど，自由度を小さくすればするほどBASICのスピードは速くしやすいのです（かといって速いBASICがユーザーフレンドリーでないというわけではありません）。また，出てくる数字はFOR～NEXTの単純ループの値を引いていないのが普通ですから，その数字すらも本当の意味では間違っています。ましてや，それぞれのベンチマークテストの合計を出したりするのはいただけません。我々は数字で示される事実に弱いだけに注意する必要があるようです。

さて，ベンチマークテストのもうひとつの目的。それは「命令やサブルーチンのスピード比較」です。これは雑誌ではあまりお目にかかれませんが，ユーザーレベルではかなり頻繁に行われていることだと思います。プログラムというものは，目的はひとつであってもいろいろなアルゴリズム，ルーチンが考えられるものです。その中でスピードが重要である場合，ベンチマークテストは欠かせません。

私もBASICを習いたての頃，マシン語などというものには目もくれず，ひたすらBASICで最適化を図ろうと頑張っていました。マシン語を勉強したのはもう少しあとのことなのですが，それもマシン語のスピードに魅力があったからではなく，純粋な知的関心からであったため，それからもBASICプログラムのスピードアップには取り組み続けていました。

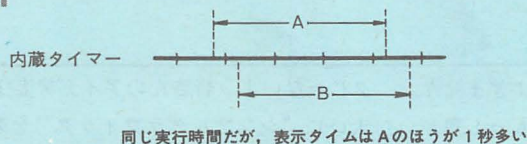
そんなわけで，何かアイデアや疑問がわくたびにベンチマークテストを繰り返していたのですが，ある日X1 HuBASICを使っていておかしなことに気付きました。同じプログラムを走らせても出てくる結果が違うのです。それがどんなプログラムであったかは覚えていませんが，そこで私が推理して得た結論は次のようなものです。

1) X1の内蔵タイマーは1個しかない。
2) TIMEの値を書き換えてもTIME$の値は変わらない。
3) 1)，2)より「TIME＝n」を実行するとタイマーの値とnの値が記憶される。
4) TIMEを読み出すときは，そのときのタイマーの値と記憶されているタイマーの値，nの値から計算して求められる。

これが正しいとすれば，同じ処理時間でも1秒の誤差（図1），同じ表示時間でも2秒近い実際の時間差があることになります（図2）。その後，特に調べてはいませんがたぶん正しいのではないかと考えています（ご存じの方がいたら教えてください）。

そのときから自分用のプログラムはリス

リスト1-A　よくあるベンチマークプログラム

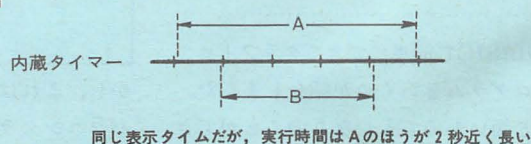
```
10 '  初 期 設 定
20 TIME=0                    '(S) TI$="000000"
30 FOR I=1 TO 回 数
40 '  処 理 内 容
50 NEXT
60 PRINT TIME                '(S) PRINT TI$
```

リスト1-B　改良したベンチマークプログラム

```
10 '  初 期 設 定
20 TIME=0                    '(S) TI$="000000"
30 IF TIME=0 THEN 30         '(S) IF TI$="000000" THEN 30
40 FOR I=1 TO 回 数
50 '  処 理 内 容
60 NEXT
70 PRINT TIME-1              '(S) PRINT VAL(RIGHT$(TI$),2)-1
```

図1　実行時間と表示タイム(1)

同じ実行時間だが，表示タイムはAのほうが1秒多い

図2　実行時間と表示タイム(2)

同じ表示タイムだが，実行時間はAのほうが2秒近く長い

ト1-Bのように改めたのはいうまでもありません。ループの時間も加わりますが，サブルーチンなどの速度比較に使うだけですからこれでよいのです。なお，命令やサブルーチンの純粋な処理時間を調べるときはFOR～NEXTの単純ループの時間を引いてリスト1-Cのようにします。

変数の魔術師

BASICプログラムの最適化のためにやっていたベンチマークテストも，前述のTIME変数のようにBASICの中身を探る手段として使われることがしばしばでした。スピードチェックを目的に始めたテストでも，途中でおもしろい発見があるとそちらのほうに没頭してしまったものです。中でもいちばん印象に残っているのは「変数とスピード」に関する実験です。

私はある日何の根拠もなしに，変数はA～Zの順に速いのではないかというとんでもない仮説を立てました。予備実験で，その差はあってもほんのわずかであることがわかっていたので，相当長い時間計らないと差が出ません。メモリダンプを見てBASICのテキストがどのへんに格納されているか知っていました。そこで，AからZまで変数名をPOKE文で書き換えながら走るという恐ろしいプログラム（リスト2）を考えついてベンチマークテストをしたのです。

結果は期待どおり，見事にA～Zの順になりました。しかし，はっきりと差が出過ぎている。変だと思い，今度はZからAの順に変数名を書き換えるようにして再テストしました。結果はZ～Aの順です。

ここで新しい仮説が浮かびました。**変数は使い始めた順に速い**というものです。考えてみれば，リスト2のプログラムではGOTOでループしても変数はクリアされないので，変数Bは2番目，Zは26番目に使われた変数であるということになり，再テストの結果でもつじつまが合います。確認のためリスト3-Aのようなプログラムを作っていろいろ実験してみました。当然のことながら変数の格納状況も調べたくなるのでメモリダンプしてみると，BASICテキストのちょっと後ろに変数名が使った順番に入っています。これですべて納得がいきました。BASICは変数を使うたびにこのテーブルの頭から参照し，未使用の変数名ならテーブルの最後に追加していくのです。リスト3-Aの50行を「50　Z＝Z＋1」とすれば26番目に使われた変数の速さがわかります。AとZの差は25個分の差ということになりますから，時間差を25で割れば1個分の差をかなり正確に知ることができます。この差はBASICによってかなり異なっていて，HuBASICは差が大きく出るようです。

さて，ここでもうひとつ仮説が浮かびます。2文字変数より1文字変数のほうが速いと考えるのが普通ですが，変数名の参照の仕組みによっては，**2文字変数のほうが1文字変数より速いこともある**のではないかということです。S-BASICではすべての変数が2文字で扱われているので基本的に関係ありませんが，なかなか興味深い話なのでS-BASICを使っている方も読んでください。

リスト3-Aをリスト3-Bのように変更して試してみましょう。HuBASICでは変数Zより速くなるはずです。このことから考えて，HuBASICでは変数名をチェックする前に変数名の字数をチェックしていると推定できます。変数Zの場合AからYまで字数と変数名の両方をチェックしてパスするのに対して，AAの場合AからZまで字数をチェックするだけで済むのでそれだけ速くなるのです。3文字以上の変数名を使った場合も同様のことがいえます。HuBASICでは1変数に対して「型」，「字数」，「変数名」の順に格納されているので，おそらくこの順番にチェックされるのでしょう。

以上の研究成果がBASICプログラムの最適化におおいに役立ったことはいうまでもありません。

関係演算とIF文

BASICでダイレクトに，

A＝2：B＝2：PRINT A＝B

とやってみてください。「－1」と表示されたでしょう。今ではすっかり当たり前となり，BASICのマニュアルにもきちんと表記されるようになった関係演算ですが，私がMZ-2000でBASICを習い始めた頃はまだポピュラーではありませんでしたので，偶然このようなタイプミスをして「－1」と表示されたときは不思議に思いました。

算術演算というのは＋，－，＊，／，＾，などの算術演算子を用いて演算を行うもので，

C＝A＋B

とすれば，A＋Bという算術式の値がCに代入されます。同様に，関係演算は等号，

リスト3-A　変数と速度(1)

```
10 A=0:B=0:C=0:D=0:E=0:F=0:G=0
12 H=0:I=0:J=0:K=0:L=0:M=0:N=0
14 O=0:P=0:Q=0:R=0:S=0:T=0:U=0
16 V=0:W=0:X=0:Y=0:Z=0
20 TIME=0
30 IF TIME=0 THEN 30
40 FOR II=1 TO 30000
50 A=A+1
60 NEXT
70 PRINT TIME-1
```

リスト3-B　変数と速度(2)

```
10 A=0:B=0:C=0:D=0:E=0:F=0:G=0
12 H=0:I=0:J=0:K=0:L=0:M=0:N=0
14 O=0:P=0:Q=0:R=0:S=0:T=0:U=0
16 V=0:W=0:X=0:Y=0:Z=0:AA=0
20 TIME=0
30 IF TIME=0 THEN 30
40 FOR II=1 TO 30000
50 AA=AA+1
60 NEXT
70 PRINT TIME-1
```

リスト1-C　時間計測用ベンチマークプログラム

```
10 '  初 期 設 定
20 TIME=0               '(S) TI$="000000"
30 IF TIME=0 THEN 30    '(S) IF TI$="000000" THEN 30
32 FOR I=1 TO 回 数
34 NEXT
36 TO=TIME:TIME=0       '(S) TO=VAL(RIGHT$(TI$),2):TI$="000000"
38 IF TIME=0 THEN 38    '(S) IF TI$="000000" THEN 38
40 FOR I=1 TO 回 数
50 '  処 理 内 容
60 NEXT
70 PRINT TIME-TO        '(S) PRINT VAL(RIGHT$(TI$),2)-TO
```

リスト2　変数速度実験

```
10 CH=65
15 ' POKE ??,CH:POKE ???,CH
20 TIME=0
30 IF TIME=0 THEN 30
40 FOR II=1 TO 100:FOR JJ=1 TO 10000
50 A=A
60 NEXT:NEXT
70 PRINT CHR$(CH);":";TIME;" ";
80 CH=CH+1:IF CH=<90 THEN 15
```

不等号の関係演算子（比較演算子）を用いて演算を行います。たとえば，

C＝（A＝B）

というのは，A＝Bという関係式の値をCに代入するものです。関係式の値は，その式が成り立つとき（真のとき）－1，成り立たないとき（偽のとき）0になりますので，上の例ではAとBが等しければC＝－1，等しくなければC＝0になるわけです。

とすれば

IF A＝B THEN～

のようになっているIF文の中身はどうなっているのでしょう。BASICマニュアルでは，

IF Ⓐ THEN Ⓑ（ELSEⒸ）

は，Ⓐが成り立てばⒷ，成り立たなければⒸを実行すると書いてありますが，内部処理としては「成り立つ」，「成り立たない」などの抽象的な表現ではなく，必ず数値で判断しているはずですね。そこで出てくるのがさっきの関係演算です。

IF（X＝＜0）＋（X＞9）THEN～

IF（Y＞0）＊（Y＝＜9）THEN～

を考えると，上の関係式は，両方成り立てば（－1）＋（－1）＝－2，一方が成りたてば－1，両方成り立たなければ0になり，THEN以降が実行されるのは－2，－1のときです。下の式では，両方成り立てば（－1）×（－1）＝1，少なくとも一方が成り立たなければ0になり，THEN以降が実行されるのは1のときです。ということは，

IF Ⓐ THEN Ⓑ（ELSE Ⓒ）

というのは，「Ⓐの値が0以外であればⒷ，0であればⒸを実行する」と言い換えたほうが明快ですね。これを使えば，

IF Z＜＞0 THEN ～

というのは，

IF Z THEN～

としたほうが余計な演算がないぶん実行速度も上がるわけです。

IF（X＞0）＊（X＜9）＋（Z＝1）THEN～

は「Xが0より大きく9より小さいか，またはZが1ならば～」という意味ですが，このとおりやってしまうと両方が成り立つとき式の値は0になり，思わぬバグの原因になったりします。皆さんもこのような経験をしたことがあるのではないでしょうか。この場合は

IF（X＜0）＊（X＜9）－（Z＝1）THEN～

としなければいけないのです。ANDやORなどの論理演算が使えるBASICでは，このようなことはありません。

さて，関係演算の意味が理解でき，カーソルの移動チェックなどで，「K＝1でかつX＞0ならばX＝X－1，K＝3でかつX＜39ならばX＝X＋1」という意味で，

X＝X－（K＝1）＊（X＞0）＋（K＝3）＊（X＜39）

などと得意げにしていたのですが，あるとき雑誌で「IF文を使うより関係演算を使ったほうが速い」と書いてあったのを見て，あまのじゃくな私は「本当だろうか」という疑問にぶつかったのです。

さっそく，リスト4-A，Bのようなプログラムを作って試してみました。10行のKに0，1，3を入れてそれぞれ試してみてください。確かに関係演算を使ったほうがすっきり書けますが，スピードは必ずしもそうではなく，むしろIF文を使うほうが速いようです。MZ-2000のS-BASICではELSEが使えませんのでリスト4-Cのようになるのですが，それでもIF文を使ったほうが1割ほど速かったのです。私は一般に**関係演算を使うほうがIF文よりも遅い**という結論に達し，以後スピードが問題になるときは必ずチェックして使うようにしました。

この差はBASICによって異なり，またアルゴリズムによっては関係演算を使ったほうがいい場合もありますので，皆さんそれぞれチェックして使いましょう。なお，AND，ORが使えるものは，リスト4-Dのように IF 文中ではこちらを使ったほうが＋，＊を使うよりも速くなります。

BASIC探検の楽しみ

こんな感じで，私はベンチマークテストとともにBASICの探検を楽しんできました。もちろん中心はプログラム作りであったわけですが，BASICへのこのような知的関心はプログラミングの大きな刺激剤であったのです。

皆さんが日頃BASICを使っていて気付いたちょっとした疑問は，大部分はマシン語の知識，ハードの知識がなくとも理解できる，少なくとも推測はできるものです。そういった疑問を疑問のままに終わらせないこと，できる限り自分の力で解決することが大切なのではないでしょうか。プログラムを作るにしろ，中身を探るにしろ，BASICそのものがアドベンチャーなのですから。

リスト4-A　関係演算を使った場合

```
10 X=20:K=0
20 TIME=0
30 IF TIME=0 THEN 30
40 FOR I=1 TO 5000
50 X=X-(K=1)*(X>0)+(K=3)*(X=<39)
60 NEXT
70 PRINT TIME-1
```

リスト4-B　IF文を使った場合(1)

```
10 X=20:K=0
20 TIME=0
30 IF TIME=0 THEN 30
40 FOR I=1 TO 5000
50 IF (K=1)*(X>0) THEN X=X-1 ELSE IF (K=3)*(X=<39) THEN X=X+1
60 NEXT
70 PRINT TIME-1
```

リスト4-C　IF文を使った場合(2)

```
10 X=20:K=0
20 TIME=0
30 IF TIME=0 THEN 30
40 FOR I=1 TO 5000
50 IF (K=1)*(X>0) THEN X=X-1:GOTO 60
55 IF (K=3)*(X=<39) THEN X=X+1
60 NEXT
70 PRINT TIME-1
```

リスト4-D　IF文を使った場合(3)

```
10 X=20:K=0
20 TIME=0
30 IF TIME=0 THEN 30
40 FOR I=1 TO 5000
50 IF K=1 AND X>0 THEN X=X-1 ELSE IF K=3 AND X=<39 THEN X=X+1
60 NEXT
70 PRINT TIME-1
```

機械へのインタフェイス

Kuwano Masahiko
桒野 雅彦

さて困ったことになりました。特集の内容がBASIC。そこに私も参加せよとの命が下ってしまったのです。どうしよう。

困ってしまったのには訳があります。私がマイコンを扱い始めたきっかけというのが，そもそもアマチュア無線仲間とのおしゃべりだったのです。

「そういえば，何でもマイクロコンピュータとかいうのがあるらしいな」

「あぁ，なんかアキバッパラに時々ある，あのでっかいICだろ」

「うん，ちっこくても一応コンピュータらしい」

「ふーん。いっちょ，作ってみっか。ここんとこ作りたいもんもないし，どうせコンピュータなんて1と0しかないんだから単純なもんだ」

「あぁ，俺も少し調べたんだけど，マッチングも取らなくていいし，全部ワイヤ配線でいいみたいだしな」

「つなぐだけ!?　それじゃプラモといっしょじゃん。それでも電気回路かよ」

「俺だって良く知らないけど，それでいいってことみたいなんだ」

「何でぇ，単純じゃん。いっちょやってみっか」と，まあ今考えれば，何ともeasyでlightな感覚で作り始めたのです。こんなところが出発点ですから，どうしてもハードウェア寄りの思考になってしまいます。ラジオや無線機を自分で設計して，部品を集め，組み立て，調整することとまったく同じ感覚でマイコンを組み立てたのですから，私にとってマイコンとは「メモリの内容に従って，デジタル入力信号を加工して出力するもの」，つまり非常にフレキシブルな論理回路であるということだけのものだったのです。

ですから，マイコンを使って何かを制御しようという，手段として見ることがほとんどで，マイコンそのものを目的物として捕らえることはまずありませんでした。

マイコンはあくまでコントローラであるとしか見ていなかったのですから。要求されるプログラムというのも単純なもので，どう頑張ったとしても256バイトを越えることはまずありませんでした。この程度のものでは，ハンドアセンブルでも何とかなりますし，何も巨大なBASICをロードする必要もないのです。さらにその後，本誌に発表されたTTLインタプリタの前身である，GAMEインタプリタとコンパイラが手に入ってからというもの，ほとんどのプログラムはGAMEとアセンブラ(i8080表記ばかり使っていた）で組んでいました。

変数は26個もあるし，四則演算やメモリアクセスも簡単にできるうえ，インタプリタが約1.5Kバイトと，かなりコンパクトであったことが，GAMEを使った理由でした。

特にMZ-80Kのように，カセットテープからロードしない限り，メモリダンプすらできないようなマイコンでは，何か事が起こる(人はそれを暴走と言う)たびにシステムからロードし直しになるのですから，大きさ（＝ロード時間）は大切でした。

私にとっては，実数演算や文字列処理などは不要でしたし，BASICでは16進も満足に扱えない（SP-5030に至って，ようやくPEEK，POKEのアドレスに16進数が使えるようになった）のですから，私のニーズにはまったく合っていなかったのです。一応BASICの記法くらいは雑誌等で眺めて覚えていたので，ブラックジャックなどのゲーム，一筆書きのパズルなどを解かせてみたり，そしてまたある時は無理数を数百桁求めてみたりと，いろいろやってはみたのですが，やはりあまり本筋という感じではありませんでした。どちらかと言えば，BASICしか動かせない世間一般のマイコンへの移植を考えなければならないときや，「情報処理」とやらの宿題をたのまれたときが主体となっていました。

BASICの基本を考える

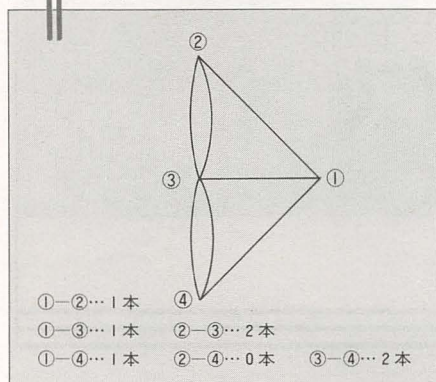
それでは，その当時私が作ったプログラムのうち，平方根を求めるもの(リスト1)と，一筆書きの解を求めるもの(リスト2)を紹介しましょう。前者は，BASICの桁数の限界（Hu BASICでも16桁）を超える精度が欲しいときに，後者は，この手のパズルをコンピュータで解かせるときの基本である，バックトラッキングの例として，ちょっとつついてみるには良い題材だと思います。アルゴリズム上でも，まだまだ高速化の道はあると思いますので，元気な人は試してみてはいかが？

1）平方根を求める

平方根に限らず，関数の微分さえできれば答が求められるニュートン法や，原始的に力づくでまともに計算していく方法など，いろいろな道がありますが，ここでは理屈を無視して，やり方だけよく知られているという，開平法と称する（らしい）方法で解かせてみることにしました。

開平法は，四則演算ができれば良いうえ，比較的楽に精度が出せるので，平方根を手計算で求める方法として，参考書等でもよく見かけるようです。

今回は，この手計算の部分をMZにやらせてみたのですが，多桁の演算をサブルーチン化したため配列のコピー等で手間を食ってしまい，速度の点では今ひとつもの足りません。100桁や200桁もの精度を要求すると画期的に遅くなりますが，勉強用と割り切って大目に見てやってください。

図 一筆書きの問題の例

2) 一筆書きを解く

いまさら説明するまでもない，大古の昔から有名なパズルです。今回はこの古典的なパズルをMZに解かせてみました。とにかく，基本的にというか泥くさくということを考え，「奇点(線が奇数本集まっている点)が3つ以上なら不可能，2つ以下なら，そこが始点」といった初歩的な知識も使いませんでした。

走らせると，点の数と，それぞれの点の間を結ぶ線の数を聞いてきますので，順次入力してください。

例に挙げたようなものならば，点の数は4，線のデータは1，1，1，2，0，2と入力します。

入力が終わるとMZが解くために使う表(点の間の道の本数の表)と，道の総本数を表示してから解きにかかります。

解いている間，何も表示させないほうが速くなるのは当然ですが，見ていてまったく面白くないので，たどっている経路を表示させるようにしました。行きつ戻りつ，解を見つけようとしている姿がわかることでしょう。

例題は答がないので少々かわいそうな気もします。延々とルートを探したあとで「impossible(不可能)」という叫びをあげますので，今度は，②と④の間に道を通してあげましょう。

BASICを利用する

さて，このような具合でBASICとは，あまりつきあっていなかった私ですが，実数やプロッタを扱うことになってハタと気が付いたのは，BASICが数値計算や周辺機器のインタフェイスとしても利用できることです。

BASICを上手にだましてやることで，アセンブラで書いたサブルーチンと周辺機器の間をとりもたせることができることに気付いてから，わりとBASICをよく使うようになりました。

このような使い方で特によく利用したのがBASICの配列変数です。整数の配列を宣言すると，BASICがその分だけメモリを開放してくれるうえ，中身を読み書きする場合に，わずらわしいPEEKやPOKEを使わず，A(I)といった具合にできてしまうというのが好ましかったのです。

配列に限らず，BASICにおける変数の宣言は，メモリを確保し，そこに名前を付けることだと考え直してみれば，よくわかるでしょう。アセンブラで書いたルーチンとの間で，変数名を使ったメモリを渡してデータの受渡しを行い，ファイルアクセスや数値計算，表示等はBASICにまかせることで，処理速度を落とさずに短期間で動くプログラムにすることができるのです。

これがさらに進むと，BASICの管理しているポインタをこっそり書き換えて……となるのですが，BASICのバージョンの違いなどであっさり動かなくなってしまうので私も最近はほとんど行いません。

このように，機械語とBASICの間のデータ引渡しの例として，メモリダンプ・プログラムを紹介しましょう。MZ-1500のBASICでは，マイクロソフト系のBASICにはほとんど付いているVARPTR(変数の格納されるアドレスを返す)という関数がないのですが，幸い，USR関数実行時に文字変数のアドレスをひとつだけ渡せるので，これを利用します。

文字変数の配列を確保して，中にスペースをつめ込んでおきます。機械語ルーチンでは，このスペースの部分をダンプリストに書き換えます。BASICは，USRから帰ってきた時点で配列の中身をPRINTします。

リスト1 平方根

```
1000 INPUT" ｹﾀｽｳ ﾊ ? ";KETA:MAX=INT(KETA/3)
1010 DIM MAIN(MAX),SUB(MAX),B0(MAX),B1(MAX),A(MAX),B(MAX)
1020 FOR I=0 TO MAX
1030  MAIN(I)=0:MAIN(I)=0:SUB(I)=0:B0(I)=0
1040 NEXT I
1050 INPUT" ﾃﾞｰﾀ ﾊ ? ";MAIN(0)
1060 '---------------------------------
1070 FOR I=0 TO MAX:B0(I)=0:NEXT
1080 CO=1
1090 FOR I=0 TO MAX:B(I)=SUB(I):NEXT
1100 ZZ=10*CO:GOSUB"MULT"
1110 ZZ=CO*CO:GOSUB"ADDS"
1120 FOR I=0 TO MAX:A(I)=MAIN(I):NEXT
1130 GOSUB"CMP"
1140 IF ZZ>0 THEN 1180
1150 FOR I=0 TO MAX:B0(I)=B(I):NEXT
1160 CO=CO+1:IF CO>100 THEN END
1170 GOTO 1090
1180 CO=CO-1
1190 FOR I=0 TO MAX:B(I)=MAIN(I):A(I)=B0(I):NEXT
1200 GOSUB"SUB"
1210 ZZ=100:GOSUB"MULT"
1220 J=0
1230 FOR I=0 TO MAX:MAIN(I)=B(I):J=J OR (MAIN(I)>0):NEXT
1240 PRINT CO;:IF J=0 THEN END
1250 FOR I=0 TO MAX:B(I)=SUB(I):NEXT
1260 ZZ=10:GOSUB"MULT"
1270 ZZ=2*CO:GOSUB"ADDS"
1280 FOR I=0 TO MAX:SUB(I)=B(I):NEXT
1290 GOTO 1070
1300 LABEL"MULT"
1310 ZC=0
1320 FOR ZI=0 TO MAX
1330  B(ZI)=B(ZI)*ZZ+ZC
1340  IF B(ZI)<1000 THEN ZC=0:GOTO 1370
1350  ZC=INT(B(ZI)/1000)
1360  B(ZI)=B(ZI) MOD 1000
1370 NEXT ZI
1380 RETURN
1390 LABEL"CMP"
1400 ZZ=0
1410 FOR ZI=MAX TO 0 STEP -1
1420  IF B(ZI)>A(ZI) THEN ZZ=1:ZI=-1:GOTO 1440
1430  IF B(ZI)<A(ZI) THEN ZZ=-1:ZI=-1:GOTO 1440
1440 NEXT ZI
1450 RETURN
1460 LABEL"SUB"
1470 ZC=0
1480 FOR ZI=0 TO MAX
1490  B(ZI)=B(ZI)-A(ZI)-ZC
1500  IF B(ZI)<0 THEN B(ZI)=B(ZI)+1000:ZC=1:GOTO 1520
1510  ZC=0
1520 NEXT ZI
1530 RETURN
1540 LABEL"ADDS"
1550 B(0)=B(0)+ZZ
```

文字変数なので，PRINT文一発で1行分の表示が行われますから，速度もモニタのダンプと比べ，それほど遅くないはずです。このPRINT文のところをQDやRAMファイルへの書き込みにすれば，何とダンプリストのファイルができてしまうことになります。

これを持って，どこかプリンタのある所に行って「ベ～ッ！」と打ち出してしまえば，ダンプリストが一丁あがり，となるわけです。

ファイル形式の共通性に注目

さて，ここからはMZ 8 bit系（Kにあらず）とはちょっと離れるのですが，DOSとBASICのファイル形式が共通である場合の便利さについて触れておきましょう。

このようなマシンとしては，PASOPIA-16やPASOPIA-1600があります。

PASOPIAの場合は，まずMS-DOSが起動し，そこからBASICをローディングして動作します。つまりBASICもMS-DOS上のひとつのファイルとして扱われているので，BASICで作ったファイルは当然のことながら，MS-DOS上のファイルとなります。

BASICからアスキーセーブを行えば，DOS上のエディタ（EDLIN等）で扱えてしまいます。

このことを使って，以前，私はBASICから，アセンブラのソースリストを作成したことがあります。高速グラフィックを行うのに，どうしても三角関数のテーブルが必要だったのですが，アセンブラにはそんな関数は持っていません。といっても，手で，DB ○○，○○，…… とたたいていたのでは，何時間かかるかわかりませんし，何より，動きがおかしいときに，テーブルがおかしいのか，プログラムが違っているのか，わからなくなってしまいます。

このときハタと気が付いたのが，BASICの存在でした。「あそこには実数演算ルーチンがある。三角関数など一発だ」ということで作ったのが，BASICでFOR～NEXTループを作り，その中でディスクに対して値をPRINTする（PRINT #1，○○といったもの）という，20行にも満たないものでした。このプログラムは，本当に短いものでしたが，非常に役に立ちました。電卓で計算して手で入力していたら何時間か

```
1560 IF B(0)>1000 THEN ZC=INT(B(0)/1000):B(0)=B(0) MOD 1000
1570 FOR ZI=1 TO MAX
1580  IF ZC=0 THEN ZI=MAX+1:GOTO 1610
1590  B(ZI)=B(ZI)+CY
1600  IF B(ZI)>1000 THEN ZC=INT(B(ZI)/1000):B(ZI)=B(ZI) MOD 1000:GOTO 1610
1610 NEXT ZI
1620 RETURN
```

リスト2　ひと筆書き

```
1000 INPUT"ﾉｰﾄﾞ ﾉ ｶｽﾞﾊ ｲｸﾂ ﾃﾞｽｶ ? ";N
1010 DIM A(N,N)
1020 L=0
1030 FOR I=1 TO N-1
1040  FOR J=I+1 TO N
1050   PRINT I;" ﾄ";J" ﾉ ｱｲﾀﾞﾉ ﾐﾁ ﾊ ﾅﾝﾎﾞﾝ ﾃﾞｽｶ ";:INPUT A(I,J)
1060   L=L+A(I,J):A(J,I)=A(I,J)
1070  NEXT J
1080 NEXT I
1090 FOR I=1 TO N
1100  PRINT
1110  FOR J=1 TO N
1120    PRINT A(I,J);
1130  NEXT J
1140 NEXT I
1150 PRINT:PRINT"TOTAL ROADS = ";L
1160 DIM W(L)
1170 S=1:D=1:C=1:W(C)=S
1180 IF A(S,D)>0 THEN 1260
1190 D=D+1:IF D<=N THEN 1180
1200   A(W(C-1),S)=A(W(C-1),S)+1
1210   A(S,W(C-1))=A(W(C-1),S)
1220   C=C-1:IF C>0 THEN D=S+1:S=W(C) ELSE S=W(1)+1:W(1)=S:D=1:C=1
1230   IF S>N THEN PRINT"IMPOSSIBLE!":END
1240   IF D<=N THEN 1180
1250 GOTO 1200
1260 A(S,D)=A(S,D)-1:A(D,S)=A(S,D)
1270 FOR I=1 TO C:PRINT W(I);:NEXT:PRINTD
1280 IF C=L THEN 1400
1290 C=C+1:W(C)=D:S=D:D=1
1300 GOTO 1180
1400 PRINT:PRINT"ANSWER:":PRINT W(1);
1410 FOR I=2 TO L:PRINT" ->";W(I);:NEXT
1420 PRINT " ->";D
```

リスト3-A　メモリダンプ・BASIC部

```
1000 '/*----------------------
1010 '  - MEMORY DUMP PROGRAM -
1020 '  ----------------------*/
1030 LIMIT$CDFF
1040 DIM A$(15)
1050 LOAD"DUMP.OBJ"
1060 GOSUB"DIMINZ"
1070 PRINT:INPUT "INPUT START ADRS = ";ADR$
1080 ADR=VAL("$"+ADR$)
1090 AL=ADR MOD 256  : IF AL<0 THEN AL=256+AL
1100 AH=INT(ADR/256) : IF AH<0 THEN AH=256+AH
1110 POKE $CE01,AL
1120 POKE $CE02,AH
1130 USR($CE00,A$(0))
1140 FOR I=0 TO 15
1150   PRINT A$(I);
1160 NEXT I
1170 GOTO 1070
1180 '..... INITIALIZE .....'
1190 LABEL"DIMINZ"
1200 FOR I=0 TO 15
1210  A$(I)=SPC(40)
1220 NEXT I
1230 RETURN
```

リスト3-B　メモリダンプ・マシン語部ソースリスト

```
 1                  ;
 2                  ; INPUT DUMP LIST TO
 3                  ;   DIMENSION OF BASIC
 4                  ;
 5 002A             WIDTH   EQU     42  ; LENGTH OF VAR
 6 0010             LINES   EQU     10H ; LENGTH OF ROW
 7 0008             CBYTE   EQU     8   ; BYTES IN 1 LINE
 8 0000             DUMMY   EQU     0   ; DUMMY DATA
 9
10 0000             ORG     0CE00H
```

かるかわからないような巨大なテーブルが1分とかからずに生成されてしまったのですから。

できあがったテーブルと，メインプログラムをエディタでつないで無事にアセンブルすることができました。

このあとしばらくの間は，友人にこの巨大なテーブルを見せて驚かせていたことは言うまでもないでしょう。BASICが他の言語用のソースを作れるというあたり，DOS上でBASICが走っていることのありがた味だなあなどと思ったものです。

当然，これとは逆に，他の言語やDOS上のエディタからBASICのソースを作ることもできます。BASICのテキストをアスキーセーブしたものは即エディタで修正できることを使って，BASICテキストを，英文ワープロを使ってきれいに整えたり，変数名の付け換え（ワープロの文字列置き換えの機能を使う）を行うなどのこともよくやりましたし，人にプログラムリストを見せるときや，リストを保存しておくときには，ワープロから印字させたりしています。ページ番号まで入るので，なかなか重宝なものです。

以上，私なりのBASICの利用法はいかがだったでしょうか。BASICを広い意味で，機械へのインタフェイスとして使うというのは，慣れてくると実に便利なものです。

まだまだ，ここで紹介しきれない利用法はいくらもあります。例を挙げて紹介すると，

(1)大型コンピュータ等に，マイコン上で作ったプログラムのソースリストを移したい場合などでは，RS-232Cでつなぎ，BASICからRS-232Cをオープンして大型コンピュータのエディタに入る文字列をPRINTします。そのあとソースリストを読み，そのままPRINTするループを作り，最後まで送ったら，今度はエディタを抜けるコマンドを送る。これで事実上，ターミナルから手でソースを入れたのと同じになります。

(2)大型コンピュータから帰ってくる結果がBASICステートメントになるようにすることで，帰ってきた結果をグラフィック表示させる。

等々……。

皆さんもアイデアをしぼって，これからBASICを活用してみてください。

```
 11 CE00 210000          LD      HL,DUMMY ; START ADDRESS
 12 CE03 3E10            LD      A,LINES   ; LINE COUNTER
 13 CE05 3284CE          LD      (LCOUNT),A
 14 CE08 D5              PUSH    DE    ; SAVE ADRS
 15 CE09 DDE1            POP     IX
 16 CE0B         SETLINE:
 17 CE0B DDE5            PUSH    IX
 18 CE0D FDE1            POP     IY
 19 CE0F CD73CE          CALL    BUFINZ ; 1 LINE CLEAR
 20              ;---- SETUP ADDRESS ----
 21 CE12 7C              LD      A,H
 22 CE13 CD50CE          CALL    BTOHEX ; BINARY TO HEX
 23 CE16 FD7300          LD      (IY),E
 24 CE19 FD7201          LD      (IY+1),D
 25 CE1C 7D              LD      A,L
 26 CE1D CD50CE          CALL    BTOHEX
 27 CE20 FD7302          LD      (IY+2),E
 28 CE23 FD7203          LD      (IY+3),D
 29
 30 CE26 110500          LD      DE,5
 31 CE29 FD19            ADD     IY,DE
 32              ;---- SETUP DATA -----
 33 CE2B 0608            LD      B,CBYTE
 34 CE2D         SETDATA:
 35 CE2D 7E              LD      A,(HL)
 36 CE2E 23              INC     HL
 37 CE2F CD50CE          CALL    BTOHEX
 38 CE32 FD7300          LD      (IY),E
 39 CE35 FD7201          LD      (IY+1),D
 40 CE38 110300          LD      DE,3
 41 CE3B FD19            ADD     IY,DE
 42 CE3D 10EE            DJNZ    SETDATA
 43              ;---- NEXT CULUM ----
 44 CE3F 112A00          LD      DE,WIDTH ; SET IX TO NEXT CULUM
 45 CE42 DD19            ADD     IX,DE
 46 CE44 3A84CE          LD      A,(LCOUNT) ; SET FINISHED?
 47 CE47 D601            SUB     1
 48 CE49 3284CE          LD      (LCOUNT),A
 49 CE4C C20BCE          JP      NZ,SETLINE
 50 CE4F C9              RET
 51
 52
 53              ;..... BINARY TO HEX CONVERT ....
 54              ; ENTRY ... ACC
 55              ; RETURN ... E=UPPER
 56              ;        ... D=LOWER
 57 CE50         BTOHEX:
 58 CE50 D9              EXX
 59 CE51 47              LD      B,A ; SAVE ACC
 60 CE52 E60F            AND     0FH
 61 CE54 CD69CE          CALL    NBLTOH ; NIBBLE TO HEX
 62 CE57 57              LD      D,A
 63 CE58 78              LD      A,B
 64 CE59 CB3F            SRL     A
 65 CE5B CB3F            SRL     A
 66 CE5D CB3F            SRL     A
 67 CE5F CB3F            SRL     A
 68 CE61 CD69CE          CALL    NBLTOH
 69 CE64 5F              LD      E,A
 70 CE65 D5              PUSH    DE
 71 CE66 D9              EXX
 72 CE67 D1              POP     DE
 73 CE68 C9              RET
 74              ;.... NIBBLE TO HEX CONVERT ....
 75              ;ENTRY ... ACC
 76              ;RETURN ... ACC
 77 CE69         NBLTOH:
 78 CE69 FE0A            CP      0AH
 79 CE6B 3003            JR      NC,NBLHI
 80 CE6D C630            ADD     A,'0'
 81 CE6F C9              RET
 82 CE70         NBLHI:
 83 CE70 C637            ADD     A,'A'-0AH
 84 CE72 C9              RET
 85
 86              ;.... CLEAR 1 LINE ....
 87              ;ENTRY ... IX=TOP OF LINE
 88              ;RETURN ... NONE
 89 CE73         BUFINZ:
 90 CE73 D9              EXX
 91 CE74 DDE5            PUSH    IX ; COPY IX TO HL'
 92 CE76 E1              POP     HL
 93 CE77 0628            LD      B,WIDTH-2
 94 CE79         BUFCLR:
 95 CE79 3620            LD      (HL),' '
 96 CE7B 23              INC     HL
 97 CE7C 10FB            DJNZ    BUFCLR
 98
 99 CE7E DD36043A        LD      (IX+4),':' ; ADRS SEPARATER
100 CE82 D9              EXX
101 CE83 C9              RET
102
103 CE84         LCOUNT: DEFS    1
```

ポケコンを使ってみよう

Takahara Hideki
高原ひでき

ちょっと趣向を変えて，ポケコンを使ってみました。BASICを通じてポケコンという機械のオペレーションに入門してみましょう。私の手元にはやや古い製品ですがシャープのスタンダード機であるPC-1500があります。

まず初めに強調しておきたいのは，パソコンとポケコンは使用目的，ひいては利用環境が大きく違います。ですから，プログラムを作る道具としては，パソコン用BASICと，ポケコン用BASICは異なるものとさえ言えるのです。

とは言っても，言語としてのBASICを見ると，ポケコン用もパソコン用もほぼ同じです。Z$を表示するときはPRINT　Z$ですし，値の入力はINPUTやA$=INKEY$，READ～DATAなどです。ループ文のFOR～NEXTもありますし，条件判断はやはりIF～THENです。SIN，LOGなどの関数の使い方もしかりです。

ここまでを基礎知識として，あとはパソコンとポケコンの使用環境の違いを考えながら，ポケコンなりのBASICによるプログラムの作り方を考えていきましょう。

表示機能の違い

まず，表示機能の違いに注目したいと思います。パソコンのプログラムを作るときは，80字×25行の画面いっぱいを使ってメッセージやデータ，計算結果を表示します。もちろん文字ごとに色を変えたり，カラーグラフィックスを使うこともあります。この画面デザインは，いまやプログラミングの重要なファクターで，プログラマの腕の見せどころのひとつでしょう。

ところが，ポケコンは基本的には1行モノクロのキャラクタ（文字専用）ディスプレイですから，デザインのしようがありません。ズバリ答がわかればいいのです。CLSやCURSORに気を配る負担がずいぶんと軽くなるわけです。試しに次のプログラムを考えてみましょう。

```
10  A＝2：B＝8
20  PRINT  A;"+";B;"=";A+B
30  PRINT  A;"×";B;"=";A*B
40  END
```

実に単純なプログラムです。これをパソコンで解くと，

```
RUN
  2＋8＝10
  2×8＝16
OK.
```

とディスプレイ上に表示されます。

では，ポケコンはどうでしょうか。ここで気がつくのは液晶ディスプレイは1行しかないのですから，パソコンのように2つの答を一度には表示できっこありません。

実際の操作手順は

RUNと ENTER を押す
　2＋8＝10　と表示される
ENTER を押す
　2×8＝16　と表示される
ENTER を押す
　何も表示されない＝終了

ポケコンは，ひとつ何か作業をしてディスプレイに表示したら，こちらの利用者側は「わかった」という気持ちをこめてENTERキーを押さなくては次に進めないようにできているのです。これは

```
10  A＝2：B＝8
20  FOR  X＝1   TO  8
30  A＝A＋B
40  PRINT  X;A;
50  NEXT  X:END
```

のように，40行でセミコロンを使った連続表示時でも同じです。

WAIT

とは言っても，毎回 ENTER キーを押すのも場合によりけり。そこで押したくないときのことも考慮されています。

＞　WAIT　（数字）

（数字）を指定すると，数字／85（秒）の間隔をあけて連続表示ができます。

したがって，ポケコンのプログラムを作る場合の力点としては，パソコンで頭を悩ます画面デザインは不要で，どういう手順で表示すれば見やすいプログラムになるかにすべてをかけるのです。いわば面積ではなく時間の制御といえます。

リストの見やすさ

もうひとつ，表示機能で見落とせないこととして，プログラムリストが見やすいということがあります。パソコンでリストを見ると，下から上にどんどんスクロールされていきますから，見たい部分を自分で選ばなくてはなりません。一度に見れる範囲は23行もあるので便利なようですが，リストを流していて「あっ，行きすぎた」という経験は，誰しもどこかで経験していることと思います。

ではポケコンは？　もちろん1行しか見れないのですが，これをカバーするために上下スクロールキー ↑，↓ があり，デバッグ中でもプログラミング中でも，自由にリストを上下にスクロールして，見たい行を見ることができるようになっているのです。これはパソコンにもぜひほしい機能ですね（turboなどにはありますが）。

計算機能

ところでポケコンは，パソコンに比べると価格が安いこともあり，機能は「ポケコンが劣っている」ように感じますが，果たしてそうでしょうか。ここでみなさんにドキッとしてもらいましょう。

例題1．1／9を9個加えると1になるはずですね。でも本当にそうでしょうか？

```
10 A=1/9:AA=0
20 FOR X=1 TO 9
30 AA=AA+A
40 NEXT X
50 PRINT "GOSA:";1-AA
60 END
```

これを確認するプログラムの例が例題1で，パソコンでもポケコンでも使えます。1－（1／9）＊9を求めて，計算の狂いを調べてみます。もちろん完璧な計算なら0になるはずなのです。0以外の数字が表示されれば，それが誤差です。まずポケコンで計算すると

　１Ｅ－10

となりました。次にパソコン(HuBASIC)で計算してみましょう。

　－４．６５６６１２９Ｅ－10

どうですか，この結果。誤差が5倍近くあり，しかもマイナスになっているのです。

これは有効数字と誤差の関係なのですが，パソコン用BASICでは，8桁以上の数字になると誤差が発生します。その点ポケコンは10桁までは大丈夫なようです。億単位の計算すらできないわけですから，パソコンはやはり個人用なんですかねえ。

ちなみに1／9を求めてみると

　1.111111111E-01　ポケコン
　.11111111　パソコン

Hu BASICは倍精度計算ができるのですが，その結果は

　.11111111111240461

と，悲惨なものです。

パソコン，ポケコンと言えども，コンピュータ＝電子計算機です。この結果は何かおかしいと感じざるをえません。とにかく巨額の取引計算やハレーすい星の軌道計算をするならポケコンで，ということになりましょうか。

メモリバックアップ機能

うっかりパソコンのスイッチを切ると，そのときメモリの中にあるプログラムやデータはすべて消えてしまいます。そういうものだと思ってしまいがちなのですが，よくよく考えてみると，ムダな話です。ところが，ポケコンはちゃんと覚えていてくれるんです。知ってましたか？

特に強調したいのは，その覚え方がハンパではなく，電源ボタンをOFFにする前と，ONにしたあとのメモリの状態がまったく同じなのです。ですからプログラムのリストだけではなく，変数すべての値までそっくりそのままなのです。

たとえば，Ａ＝３とキーインして，電源ボタンをOFFにし，再びONにするとどうなるか？

　A　[ENTER]
　　3

とちゃんと表示されます。プログラムの途中で [BREAK] キーを押して，電源ボタンをOFFにしたときも，どこで中断したかを覚えていますからスイッチONのあと，

　CONT　[ENTER]

で再開できます。

この理由は，ポケコンではCMOSという低消費電力型のメモリ（RAM）を使い，そこに書かれた情報をいつも電池でバックアップしているからです。メモリのすべての内容が，電源ボタンのON／OFFにまったく依存しないなんて，素晴しいと思いませんか。

例題2．お買い物メモ

スーパーで買い物するさいに，持ち合わせが少ないと不安に襲われますね。そこでポケコンを使ってみましょう。

```
10 G=0
20 INPUT "? KINGAKU=";A
30 INPUT "? KOSU=";K
40 PRINT "SHOKEI=";A*K
50 INPUT "OK? (Y/N)";Z$
60 IF Z$="Y" THEN 90
70 IF Z$="N" THEN 20
80 GOTO 50
90 G=G+A*K
100 PRINT "GOKEI=";G
110 GOTO 20
```

ここでは，ちょっと変わったプログラムにしてみました。プログラムはRUNで起動すると，変数の内容がすべてクリア（数字は0，文字列は“”）されますが，途中から始めると，保持されています。

そこであらかじめ [F1] に

　GOTO　20

と登録しておきます。買い物を始めるときだけ [F6] のRUNを押し，次からは [F1] を押して使います。中断するときはいつでも [OFF] ボタンを押せばいいのです。

さらに便利なことには，時刻を表す変数TIMEも電源のON／OFFと無関係なのです。メモリの内容保持機能とこの時計機能を連動すれば素晴しいソフトが作れそうです。

そして携帯性

そして決定的にパソコンと違うのは，パソコンが机の上に置いて使うようにできているのに対し，ポケコンはどこでも使えるような携帯用であることです。

いきおいプログラムも，本来はこの携帯性を考慮したものでなくてはならないはずです。では携帯性を考慮したプログラムとはどんなものでしょうか。それは人それぞれで違うはずです。セールスマンの人がセールスデータの収集，加工分析に外出先で，銀行などではどこにでも持ち運べる利率の計算機として窓口で使っています。また大学では，狭い実験室の中で場所をとらずにすぐ複雑な計算をこなすコンピュータとして使うケースが多いようです。みなさんも自分の生活を眺めてみて，自分なりのポケコンの使い方を探してみてください。とりあえずここでは私が使ってみて便利だと感じたことがある自作プログラムを2つ紹介しておきます。

例題3．〇×テスト回答用紙

週刊誌の性格分析から運転免許のペーパーテストまでたくさんの〇×式テストがありますが，ポケコンを回答用紙兼採点機として使いましょう。

あっ，ばれましたか。以前BASIC塾でサンプルプログラムに使ったものです。でも

このプログラムは私自身PC-1500用が実用的だったので掲載したんです。

使い方は問題数を入力して，問1から順に回答を記入します。○なら [O]（ゼロ），×なら [+]（プラス）です。入力ミスしたら [B] で戻ります。終了後，正解を記入します。あとで点数が表示されたあと，間違った問題番号をまとめて表示する親切設計です。

例題4．ゴルフコンペ順位計算機
小規模なゴルフコンペならポケコンで順位とスコアを計算しちゃいましょう。もちろんプリンタつきならベターです。

ゴルフコンペのあと，成績を幹事氏が電卓で計算している姿をよく見かけるので試しに作ってみたのです。使い方は510行から998行までの間でDATA文として名前，ハンデ，アウトのスコア，インのスコアの順に語入します。そしてRUN するだけです。記入例を510行から540行までに納めています。なお名前を“”でくくるのを忘れずに。

★　★

どうでしたか。まだポケコンがそう小さくないので，本格的に個人で使っている人は少ないようですが，近い将来いまの電卓ぐらいにはなるはずです。その日を夢みてポケコンをたまには使ってみてください。

例題3．○×テスト

```
10 P=0:WAIT 200
20 INPUT"> HOW MANY Q.:";N
30 DIM Y(N),A(N)
40 PRINT "O=<O:ZERO> X=<+> BACK=<B>"
50 WAIT 30
60 FOR X=1 TO N
70 PRINT "> Q.";X;"...";
80 GOSUB 300
90 IF K<>3 THEN 100
95 X=X-1:CLS:GOTO 70
100 Y(X)=K
110 NEXT X
150 FOR X=1 TO N
160 PRINT "ANS";X;"...";
170 GOSUB 300
180 IF K<>3 THEN 190
185 X=X-1:CLS:GOTO 160
190 A(X)=K
200 IF K=Y(X)THEN PRINT"OK.":P=P+1:GOTO220
210 PRINT"MISTAKE!"
220 NEXT X:WAIT 100
230 PRINT "POINT:";INT(P/N*100+.5)
240 PRINT "MISS...";
250 FOR X=1 TO N
260 IF A(X)<>Y(X) THEN PRINTX;
270 NEXT X:END
300 REM --- SUB ---
310 GET K$
320 IF K$="" THEN 310
330 K=0
340 IF K$<>"O" THEN 360
350 K=1:PRINT"O":RETURN
360 IF K$<>"+" THEN 380
370 K=2:PRINT"X":RETURN
380 IF K$<>"B" THEN 310
390 IF X=1 THEN 310
400 K=3:RETURN
```

例題4．ゴルフプログラム

```
10 M=0
20 READ Z$,A,B,C
30 IF Z$="*" THEN 50
40 M=M+1:GOTO 20
50 RESTORE 500
60 DIM N$(M),H(M),N(M),O(M),I(M)
70 FOR X=1 TO M
80 READ N$(X),H(X),O(X),I(X)
90 N(X)=O(X)+I(X)-H(X)
100 NEXT X
110 FOR Y=1 TO M-1
120 FOR X=Y+1 TO M
130 IF N(X)>N(Y) THEN 190
140 Z$=N$(X):N$(X)=N$(Y):N$(Y)=Z$
150 Z=O(X):O(X)=O(Y):O(Y)=Z
160 Z=I(X):I(X)=I(Y):I(Y)=Z
170 Z=H(X):H(X)=H(Y):H(Y)=Z
180 Z=N(X):N(X)=N(Y):N(Y)=Z
190 NEXT X:NEXT Y
200 FOR X=M TO 1 STEP -1
210 PRINT X;N$(X);":NET";N(X)
220 PRINT O(X);"-O";I(X);"-I";H(X);"-H"
230 NEXT X
240 END
500 REM DATA "NAME",H,O,I
510 DATA "Q-TARO",36,59,69
520 DATA "PAMAN",6,38,41
530 DATA "HATTORI",12,42,49
540 DATA "DORA",18,51,62
999 DATA "*",0,0,0
```

求む究極のBASIC

Iwai Ippei
祝　一平

BASICという言語はかわいそうな言語である。初心者でも使えるという理由によって大勢の人に使われるようになったが，その反面，初心者用だからという理由でまともに扱ってもらえなくなってしまった。つまり，研究対象とはなりえず「進化」が止まってしまったのだ。グラフィック関係の命令を取り込んだりして高機能・多機能化はしているが，もっとも基本的なアルゴリズムの表現法などは貧弱なままに取り残されてしまっているのだ。

たとえばIF文である。

「IF～THEN～ELSE」が使えるのはよいが，バカバカしいことに1行中に書かなければならない。だから，時として書ききれなくなってしまい，GOTOを使って分岐をしなければならなくなるのである。

もし1行中に書ききれたとしても，べったりとした「詰め書き」になりがちで，80字を越えて画面の右端から左端へ折り返したりすると，読みにくさは頂点に達してしまうのである。ぎーっ！

そこでちょいとあたりを見渡せば，解決策は江ノ島の海水浴客のようにいくらでもころがっているのである。たとえば，

IF～THEN〔～〕ELSE〔～〕END IF

のようにカギカッコも付けられるようにしてしまうのである。すなわち，「カギカッコ入りのIF文は，複数行にわたる可能性もあるんだぞよ」である。MZ-2500のBASICには同じような機能があるが，どっこいこちらはちゃんとしたブロック文の概念をBASICに持ち込んだ点が違うのだ。MZ-2500のほうはブロック文という考え方がないから，使いごこちはあんまり良くない，と私は独断するのである。

私はこの小文中で，このよーにしてBASICの明日を考えつつ，日ごろのイライラを解消したいと思うしだいであった。

というわけで

まずは，最初に言ったようにBASICにもブロック文という考えを取り込んでしまうのである。これは今のところIF文以外ではあんまり使い道がない感じだけど，BASICの近代化および明るい未来には必要なものと確信するのである。

次に考えるのはサブルーチンコールの拡張である。現在のBASICではまともなサブルーチンコールができない。どういうことかというと，GOSUBには次のような欠点があるのだ。

①引数（パラメータ）を記述できない。
②サブルーチン内で局所変数を使えない。
③関数を記述できない。

具体的に①は，

```
10 A=1:B=5:GOSUB 100
```

のように，変数に値を入れてGOSUBしなければならないということである。これは最低である。

②は飛び先でうかつに変数をいじれないということである。たとえばFOR文ではIという変数がよく使われる。そこで，

```
10 FOR I=1 TO 100
20 GOSUB 100:NEXT
   :
```

というのは，どこにでもころがっているプログラムであろう。そこでついつい飛び先のサブルーチンでも

```
100 FOR I=10 TO 50
```

などとやってしまい，「れれれのれ」ということになってしまう。プログラムが大きくなると，この「変数管理」は白熱してきてバグ取りの効率を下げることになるのである。

③はよーするにDEFFNで記述できない関数のことを言っているのである。DEFFNは1行で記述できる「簡単な式」しか許されないのだ。だから，

・Nが素数なら1，それ以外は0を返す
・A＄の内容を逆順にしたものを返す（たとえば“BOOK”→“KOOB”）

というような関数は記述できないのである。以上の点から正しいサブルーチンコールは次のようになるべきである。

a)　GOSUBはそのまま残す。
b)　「PROC SUB1(X, Y)」という。PROC(PROCEDURE：プロシージャ：手続き)文を作り

前号までのあらすじ

東西新聞社の創立百周年記念事業のひとつである，山岡士郎と栗田ゆう子の「究極のメニュー」作りと時を合わせて，私・祝一平は「究極のパソコン：満開一号」作りに取り組んでいた。しかし私はやがて大きな壁に「こんにちは」してしまったのである。それは標準搭載の言語をどーするかということであった。とにかくもBASICは載せざるをえない。しかし私はBASICがあまりにも貧弱，ノロい，イモい，サバい，その場しのぎ，ドラ1バンバンのハネ満であることを痛感していたのであった。そんなある日，私が自宅のドアを開けるとハガキが落ちていた。それには「この家のどこかに宝が隠されている」とは書いてなく，その代わりに「悪いシステムプログラマに誘拐されてしまいました。助けてください。BASICより」と書いてあったのである。私は「ざけんじゃない！　オレは妖怪探偵ちまちまじゃねえ。こんなハガキを出すヒマがあるなら，さっさと逃げりゃいーじゃねーか」とつぶやき，憎憎しげにハガキを破り裂こうとしたのである。しかし私は寸前で思いとどまった。それは“悪いシステムプログラマ”という言葉にひっかかったからである。やがて，さまざまな点と線がゆっくりと流れていき，ついにひとつの姿にまとまっていった。私は戦いの始まりを予感しつつ，近所の酒屋で買ったカンビールを冷蔵庫の前に置いたのであった……。

10 SUB1(5, 6)

とするだけで飛んでくるようにする。もちろんSUB1が定義されてなければエラーになる。リターンはRETPROCである。

c) 「FUNC KANSU1(X, Y)」という，FUNC(FUNCTION：ファンクション：関数) 文も作る。これは関数定義である。返す値は，「RETFUNC～」で指定する。

d) それぞれのPROC, FUNC中で使う変数は，当然局所変数で，なおかつ「宣言」しなければならない。

最後のd)は，ちょっとナニだから，

d′) それぞれのPROC, FUNC中で使う変数は自動的に関数名を頭につける。たとえばSUB1というPROCの中のXという関数はSUB1_X という変数として扱われる（“_”アンダーバーをはさんでいるのがミソ。このアンダーバーは普通は使えないようにする)。こうすれば外部から参照，書き込みもできる。SUB1(X, Y)で受けた引数も，それぞれ（広域変数では)SUB1_X, SUB1_Yとなっている。この場合は再帰呼び出しなどができなくなるがしかたない。

方法としては，d) とd′)の中間として，「PROC」と「SPECIAL PROC」の2つを作り，SPECIALのほうはd) に書いたようなやつにするという手もある。よしよし，これをd″)としよう。そこでさっそく，リスト1が「ブロック文とPROC, FUNC文を使ったサンプルプログラム」である。いわずもがなだが，PROC文にはBASICのコマンドを自由に増やすような効果があるのだ。おっと，念のために言っておくが，PROCとFUNCは，値を返すかどーかの違いだけである。

次に何をやり玉にあげよーかなと捜してみると，あったあった。変数タイプである。まず，文字列の長さが255文字以下という制限をとっぱらう。ここは最低でも65535文字まで許してしまおう。いやいや，やはり32ビット(＝4バイト)で &HFFFFFFFF≒4096Mバイトまでの長さを許すことにしよう。当然整数型も現在の倍の32ビットにしてしまう。実数型と倍精度型はそのままにしておいて，なおかつ倍倍精度型も作ってしまう。う～ん，まだなにか足りないなー。

というわけで，「可変精度実数型」というのを作ってしまえ。一般的に実数の精度は仮数部の桁数（範囲）と指数部の桁数（範囲）で決まるので，

DEF XPR(仮数部桁数, 指数部桁数) A－F

などとしてしまう。XPRというのは

X PRECISION（エックス精度）

というコジツケである。たとえば，

DEF XPR(100, 10) A－F

とすると

$$\underbrace{-0.877\cdots\cdots5}_{100桁}\times10^{\underbrace{-123456789}_{10桁}}$$

という数字も扱えるようになってしまうのだ。計算はBCD（2進化10進数）で行うことにする。もちろん，精度が高ければ演算速度は遅くなるしメモリも食うが，それはそれでよいのだ。おっと，いちいち精度を指定するのは面倒だから，

XPR1→XPR(20, 10)

XPR2→XPR(50, 10)

ぐらいはあらかじめ用意しておくことにする。

次に「むっ」とするのはファイル関係である。たとえばファイル名は表示できるが「ファイル名を返す関数」が存在しない。よって，

A$＝FILENAME$(“0:”, 5)

のように，「n番目のファイル名」を返す関数を作ってしまう。なければヌルストリングを返す。それから，現在のファイル名指定には「ワイルドカード」が使えないのは最低といわざるをえない。よって，

FILES “＊.Bas”

とすれば“～.Bas”のファイルだけを表示するようにしてしまう。さらにもう少しひねって，

FILES ,BIN（もしくはbin）

とするとすべての機械語ファイルを表示するようにする。アスキーファイルの場合は「ASC」だな。もちろん，ワイルドカードとの併用も可である。まだまだあるぞ。プログラムサイズを表示する場合は

FILES ,, SIZE

である。

そいうわけで

残念ながら誌面が尽きかけてきた。若干もの足りないがここいらへんでやめておくのである。なんだかんだ言っても，私が書いたことは基本的にはほかの言語，システムではすでに実用化されていることなのである。だから，私でなくても思いつけることしかあえて書いていないつもりである。

BASICの未来を考えると完全にまっ暗である。たとえばCPUがZ80から8086に変わっても，基本的には何も進化しなかった。8ビット用のBASICがメモリや速度の点で大きな制約を受けていたということから考えれば，CPUが16ビットになった段階で何らかの進化があって当然だったはずなのだ。しかし，世のシステムプログラマの方々にはそんなことは思いもよらなかったらしく，「機能が制限された8ビット用BASIC」の影をいつまでもずるずると引きずっている。BASICは初心者用言語である。だからこそ，ほかの言語よりも重要なのである。まもなくホビー用として16ビットマシンが，ひいては32ビットマシンまで広まるであろう。そのときにどんなBASICが出てくるかを私は手ぐすね引いて待っているのだ。

BASICまだまだ可能性を秘めた言語なのである。

リスト1 サンプルプログラム

```
100 IF   [ ((A=0) OR (B=0))
110         AND (DEVICE$="EMM1:") ]
120 THEN [ GOSUB "INIT-DEVICE":              :'これも許される
130         DUMP-MAP(DEVICE$,0,15):]         :'PROCを呼びだす
140 ELSE [ PRINT "桃から産まれた桃太郎":
150         PRINT "許さん！",
160               KREVERSE("るすすを血き生"):] :'これは関数
170 'ENDIF は省略できるのだ
180 END
190 '
200 PROC DUMP-MAP(DEV$,START,LAST)
210 FOR I=START TO LAST                :'この LAST は DUMP-MAP_LAST
220      '省略
230 NEXT
240 RETPROC
250 '
260 FUNC KREVERSE(K$)                  :'漢字を逆順にする関数
270 R$="":LAST=LEN(K$)/2               :'この LAST は KREVERSE_LAST
280 FOR I=LAST TO 1 STEP -1
290     R$=R$+MID$(K$,I*2-1,2)
300 NEXT
310 RETFUNC R$
```

Oh！MZを音読しよう

専門誌の正しい読み方

Koichi Tetsuji
挙市 哲司

コンピュータ用語には英単語やその省略形がいっぱいです。これらを正しく発音することは用語の理解につながります。AUTOは「アウト」ではなく「オート」と読んでいただきたい。これは，円滑なコミュニケーションのための「音読講座」です。

皆さんはOh！MZを読むときにちゃんと声に出して読んでいますか。周りの人がおや？　と振り向いたところであの表紙を見せる。ここまでしてこそ真の（その筋の）読者です。さて，そのときに間違った読み方をして恥をかくことのないように，これから正しいコンピュータ用語の読み方というのを説明してみたいと思います。

読み方なんて自分でわかっていればどうでもいいじゃないか，と言う人もいるかもしれませんが，とんでもありません。日本語を自分勝手な読み方で読んでいる人がいるでしょうか。ちゃんと読んで話しているからこそ，人に言葉すなわち情報を伝えて，その情報を理解してもらうことができるのです。

正しく読むことの効果は，人に理解してもらうだけではありません。自分が理解する上でもたいへん重要です。文字はただ並んでいるだけではなく，そこには意味があります。正しく読むことによってその意味を理解できるのです。

驚かすのはこれくらいにして読み方講座に入りましょう。今回はBASIC特集の一環ですのでBASIC用語が中心です。受講料480円はもうお支払いいただいたことと思いますが，まだの方はすぐにレジへどうぞ。

コンピュータ用語は英語である

コンピュータのハードやソフトはこれまで主にアメリカで発達してきました。したがってその用語には英語（あるいはその省略形）が使われています。当然といえば当然です。つまり，コンピュータ用語にはなんらかの意味があり，単なるアルファベットの羅列ではないということです。たとえばZ80マシン語のLD命令，これは「エルディー」ではなく「LoaD（ロード）」であり，「転送する」という意味です。これだけでも正しく読むということの重要さがおわかりいただけるかと思います。アセンブリ言語に限らずすべてのコンピュータ用語は，正しく読むことがその意味を理解するためのいちばんの近道です。

英語と聞いてビビル人はOh！（これもそう）MZの読者にはいないと思いますが，英語といってもそう難しいものではありませんからご安心を。できれば英和辞典を片手にどうぞ。

BASICは単語のオンパレード

それではBASIC用語の読み方に入りましょう。BASICは初心者用の名のとおり，簡単な英単語やその省略形で構成されています。表1に主なBASICの命令，表2に関数について読み方などをアルファベット順にまとめてみました。これでBASICプログラムは自由自在に読めるように練習しましょう。元の英語の意味とBASICにおける意味を比べてみるのもおもしろいと思います。

あと，補足する点が少しあります。まず，プリンタ関係の命令についてですが，S-BASICではプリンタに出力する場合，命令の後ろに/Pを付けます。このPはプリンタ（Printer）のPです。HuBASICでは命令の頭にLですね。これはラインプリンタ（Line Printer）のLです。それぞれ，……スラッシュピー，エル……と読んでください。それから，S-BASICのファイル関係の命令にROPEN，WOPEN，VERIFYがありま

Oh！MZは難しいよ〜

●初心者にはわけのわからない言葉が多すぎて困っています。助けてくれ〜。
城越　明人（16）福井県

●もっと説明などをわかりやすくしてほしい。意味のわからない言葉がある。たとえばCP/M。
波多野　良博（15）新潟県

●まだパソコンを始めて1年ぐらいの初心者のため，説明が難しいところやわからない言葉が多い。用語の解説集のようなものを載せてほしい。
鈴木　祐一（25）神奈川県

●私は今年パソコンにさわったばかりのパソコン1年生です。Oh！MZは私には少し難しいみたいです。1ページか2ページでよいから，1年生のページを設けてほしい。わかりやすくBASIC，マシン語を解説してほしいと思います。
安藤　紀之（40）北海道

●しだいにマシン語レベルの内容が増えてきて難しくなってきているので，なかなか自分の能力が雑誌に追いつけなくなってきている。もう少しやさしい内容でお願いしたい。
半澤　裕子（26）宮城県

●初心者の僕にはほとんど理解できない内容ばかりだけど，いつか理解できる日がくると信じて毎月買ってます。
加納　正則（16）岐阜県

●もし12月号のハード特集が難しかったら……いやです。
島野　嘉樹（16）兵庫県

●私は英語が苦手です！　ただそれだけです。
長谷川　浩（16）神奈川県

●S-OSって何ですか（あっ見放さないで！）どうやって手に入れるの？
椿井　健司（15）愛媛県

●STUDIO MZに出てくる「㈱満開製作所」というのは何のことですか？
坂本　真志（14）島根県

●「PSGはてりめぇである」の「てりめぇ」とは何ですか？
塩田　耕司（16）徳島県

●PITMANの解答，とりあえず19，25，26面教えてぇ〜。
松下　耕三（16）大阪府

リスト1

```
10 FOR I=32 TO 255
20 PRINT CHR$(I);
30 NEXT I
40 END
```

すが，それぞれリードオープン，ライトオープン，ベリファイと読みます。

それではリスト1を見てください。これはアスキーコード32から255までのキャラクタを表示するプログラムです。これを読んでみましょう。

10行　フォア　アイ　イコール　32　トゥ　255

20行　プリント　キャラクタダラー　かっこ　アイ　かっこ閉じる　セミコロン

30行　ネクスト　アイ

40行　エンド

さて，ここで＝や$(　)；などのキャラクタが出てきました。リスト1のプログラムを実行させるともっといろいろなキャラクタが表示されます。全部読むことができますか？　そこで，お次はキャラクタ編といきましょう。

キャラクタにも読み方がある

キャラクタを全部正確に読める人はあまりいないのではないかと思います。たとえば，*という記号にしてもBASICでは掛け算の記号ですが，キャラクタとしての名前は星印またはアスタリスクですね。

JIS（日本工業規格）ではこれらのキャラクタの名前がすべて決まっています。しかし，読み方としてそれに忠実に従う必要はないでしょう。誰が聞いてもそれとわかる読み方であればいいのではないでしょうか。ところが，仲間内ではわかっていてもほかの場所では全然通用しなかったということが多々あります。そこで，通称とJISの読み方，好ましくない読み方を表3にまとめてみました。

どうですか？　へえ，こんなふうに読むのか，知らなかった，というのがけっこうたくさんあるのではないでしょうか。全部覚える必要はありませんが，間違った読み方をしていたという人は直してくださいね。

表にはまだ空欄がありますが，これは皆さんで埋めましょう。アンケートはがきの片隅にでも書いて送ってください。

表1　BASICの主な命令の読み方

命令	読み方	元の英語	英語の意味
AUTO	オート	automate	自動化する
BOX	ボックス	box	箱
CIRCLE	サークル	circle	円
CLEAR, CLR	クリア	clear	片付ける
CLOSE	クローズ	close	閉じる
CLS	クリアスクリーン，シーエルエス	clear screen	画面消去
COLOR	カラー	color	色
CONSOLE	コンソール	console	操作卓
CONT	コント，コンティニュー	continue	続ける
CURSOR	カーソル	cursor	(計算尺の) 滑子
DEF FN	デファインファンクション，デフエフエヌ	define function	関数定義
DELETE	デリート	delete	削除する
DIM	ディメンジョン	dimension	寸法，次元
DIR	ディレクトリ，ディーアイアール	directory	住所氏名録
END	エンド	end	終わり
FILES	ファイルズ	files	書類さし
FOR～NEXT	フォア～ネクスト	for～next	向かって～次
GET	ゲット	get	取る
GOSUB	ゴーサブ	go subroutine	サブルーチンへ行く
GOTO	ゴートゥ	go to	～へ行く
IF～THEN～ELSE	イフ～ゼン～エルス	if～then～else	もし～それなら～代わりに
INP	インプット，インプ	input	入力する
INPUT	インプット	input	入力する
KEY	キー	key	鍵
LABEL	ラベル	label	ラベル，貼り紙
LIMIT	リミット	limit	境界
LINE	ライン	line	線
LIST	リスト	list	表
LOAD	ロード	load	荷物を積む
LOCATE	ロケート	locate	位置を示す
MON	モニタ	monitor	監視装置
MUSIC	ミュージック	music	音楽
NEW	ニュー	new	新しい
ON	オン	on	～に基づいて
OPEN	オープン	open	開く
OUT	アウト	output	出力
PAINT	ペイント	paint	塗る
PLAY	プレイ	play	演奏する
POKE	ポーク	poke	突っ込む
PRESET	ポイントリセット，ピーリセット	point reset	点を置き直す
PRINT	プリント	print	印刷する
PSET	ポイントセット，ピーセット	point set	点を置く
READ～DATA	リード～データ	read～data	読む～情報
REM	レム（リマーク）	remark	注意
RENUM	リナンバ	renumber	番号を付け直す
RESTORE	リストア	restore	元どおりにする
RESUME	リジューム	resume	再び始める
RETURN	リターン	return	戻る
RUN	ラン	run	走らせる
SAVE	セーブ	save	蓄える，取っておく
SCREEN	スクリーン	screen	映像画
STOP	ストップ	stop	止める
TRON/OFF	トレース　オン/オフ	trace on/off	追跡する/やめる
USING	ユージング	using	使いながら
USR	ユーザー	user	使用者

Oh!MZは難しいのだだだ……

とりあえず，どーしてOh!MZが難しいのかを説明するのである。

第一に用語が難しいというわけだ。しかし，これは「難しい」以前の問題なのである。なぜなら，用語を憶えなければどーにもしょーがないわけで，早い話が難しいなんちうぜーたくなシロモノではなく，金色に光り輝く「必修」なのである。

次に，内容が難しいというわけだ。しかし，これもおそらく「用語がわからないから内容もわかんなーい」という展開が多いと見る私であった。たとえば「リロケータブル」という用語などなどを知らなければ，今月のSENTINELの中のALLOCATERやBOUNCERの説明もわかんないわけである。「用語は全部わかるけど内容が難しくてわかんない」という人は，そうはいないはずなのだ。

このよーにゴリ押ししていくと，「まずは用語を知ることだ」ということになるのである。しかしそれでもまだわかんない人がいるだろう。そこで私は突然逆上するのである。

てめーら，噛んで含むような記事なんか期待すんじゃねー。ババンバババンバンバン毎月買ってるか！　なのだ。毎月用語解説を載せれるわきゃねーだろ。そんなことだから最近の若いモンは，骨が弱かったり，鉛筆を削れなかったり，魚をオロセなかったり，朝礼で倒れたりするんでい。Oh!MZは難しいから価値があるんだ。頭のてっぺんから尻尾の先まで，すんなり読んじまえるよーな軟弱な雑誌なんか，価値がねーじゃねーか。難しさを乗り越えてこそ向上っつうもんがあるんでい。苦しいからこそRPGは面白いんでい。Oh!MZはドラゴンなんだっ。わかったかかか！　　（祝　一平）

Let's Communicate!

さて，これまでの説明でBASICについては完璧に読めるようになったことと思います。人に伝えるだけなら別にアルファベットをそのまま読んでも相手に通じることは通じます。しかし，それはアルファベットを並べて会話しようとするようなもので，相手はいったん書きとめでもしない限りその意味を理解することはできません。

編集室に質問電話をかけてくる人の中で，変な読み方をして担当の人相手に漫才をふっかける人がいるそうですが，電話では読むしか伝える方法がありませんから，きちんと読まないと相手に質問の内容を理解してもらえません。今後質問電話をする方はこの講座をよく読んでこの講座の読み方に統一するようにすると，円滑なコミュニケーションができるのではないでしょうか。

ついでに，キーボードのキーの呼び方についても表4にまとめておきます。よく，SHIFT+，CTRL+，GRAPH+という表現がありますが，これはそれぞれシフトキー，コントロールキー，グラフキーを押しながら指定されたキーを押すという意味で，シフトプラス，コントロールプラス，グラフプラスと読みましょう。

本講座ではBASIC中心に解説してきましたが，BASICに限らず，マシン語やそのほかの言語においてもマスターしようとするときは，読み方も一緒に覚えましょう。それが上達へのいちばんの近道なのです。

表4　キーボードの読み方

キートップ	呼び方
CR, ↵, ENT	リターン
BREAK	ブレイク
DEL	デリート
INST	インサート
HOME	ホーム
CLR	クリア
↑	カーソルアップ
↓	カーソルダウン
→	カーソルライト
←	カーソルレフト
SHIFT	シフト
CTRL	コントロール
GRAPH	グラフ，グラフィック
英数	エイスウ
カナ	カナ
CAPS LOCK	キャピタル（ス）ロック
ESC	エスケープ
(H)TAB	（ホリゾンタル）タブ
XFER	エックスファー
F1～F10	ファンクションキー1～10，エフ1～エフ10
ROLL UP	ロールアップ
ROLL DOWN	ロールダウン
HELP	ヘルプ
COPY	コピー

表2　BASICの関数の読み方（発音しにくいものは英字をそのまま読んでもよい）

関数	読み方	元の英語	英語の意味
ABS	アブソリュート	absolute	絶対の
AND	アンド	and	かつ
ASC	アスキー	ascii	アスキー
ATN	アークタンジェント	arctangent	逆正接
CHR	キャラクタ	character	文字，記号
COS	コサイン	cosine	余弦
EOF	エンドオブファイル	end of file	ファイルの終わり
EXP	イクスポーネント	exponent	指数
FAC	ファクトリアル	factorial	階乗
FIX	フィックス	fix	固定する
FRE	フリー	free	自由な
HEX	ヘックス，ヘキサ	hexadecimal	16進数の
INKEY	インキー	input key	キー入力
INT	インテジャー	integer	整数
LEFT	レフト	left	左
LEN	レングス	length	長さ
LN	ログ　ナチュラル	logarithm natural	自然対数
LOG	ログ	logarithm	対数
MID	ミドル，ミッド	middle	中間の
NOT	ノット	not	～でない
OR	オア	or	あるいは
PAI	パイ	pi	円周率
PEEK	ピーク	peek	のぞく
POINT	ポイント	point	位置
RAD	ラジアン	radian	ラジアン
RIGHT	ライト	right	右
RND	ランダム	random	でたらめの
SGN	サイン	sign	符号
SIN	サイン	sine	正弦
SPC, SPACE	スペース	space	空間
SQR	スクウェアルート	square root	平方根
STR	ストリング	string	列
TAB	タブ	tabulate	表にする
TAN	タンジェント	tangent	正接
TI, TIME	タイム	time	時間
VAL	バリュー	value	数値
XOR	イクスクルーシブオア	exclusive or	排他的論理和

表3　記号の読み方

	通称	JIS名称	好ましくない読み方
	スペース	間隔	空白
!	エクスクラメーションマーク，ビックリマーク	感嘆記号	オッタマゲーションマーク
"	ダブルクォーテーション(マーク)	引用記号，ウムラウト	上ちょんちょん
#	シャープ，ナンバー，井桁	番号記号	
$	ダラー	通貨記号ドル	エス棒
%	パーセント	パーセント	
&	アンパサンド，アンド	アンド記号	
'	シングルクォーテーション(マーク)	アポストロフィ，アクサンテギュ	上ちょん
(	カッコ	左小括弧	
)	カッコ閉じ(る)	右小括弧	
*	アスタリスク	星印記号，乗算記号	掛ける(演算内では可)
+	プラス	正記号，加算記号	足す（演算内では可）
,	コンマ	コンマ	下ちょん
−	マイナス，ハイフン	負記号，減算記号，ハイフン	棒，引く(演算内では可)
.	ピリオド	ピリオド	てん
/	スラッシュ	斜線，除算記号	ななめ，割る(演算内では可)
:	コロン	コロン	ちょんちょん，(たて)てんてん
;	セミコロン	セミコロン	ちょんコンマ，(たて)てんちょん
<	小なり	不等号（　より小）	
=	イコール	等号	わ
>	大なり	不等号（　より大）	
?	クエスチョンマーク	疑問符号	はてなマーク
@	アットマーク	単価符号	エーマル，アルファ
[	大カッコ	左大括弧	
¥	円，円マーク	通貨記号円	
＼	バックスラッシュ		逆ななめ
]	大カッコ閉じ(る)	右大括弧	
^	アッパーアロー	上向き矢印記号，シルコンフレックスアクセント記号	うえ山
_	アンダーライン，アンダーバー	アンダライン	下棒
`	アクサングラーブ	アクサングラーブ	逆上ちょん
{	中カッコ	左中括弧	
|	パイプ	分離記号，論理和記号	たて棒
}	中カッコ閉じ(る)	右中括弧	
‾	オーバーライン，オーバーバー	オーバライン	上棒
£	ポンド	通貨記号ポンド	エル棒
。	句点	句点	まる
、	読点	読点	てん
「	カギカッコ	始括弧	
」	カギカッコ閉じ(る)	終括弧	
・	中黒	中点	てん
ー	長音記号	長音記号	
゛	濁点	濁点	てんてん
゜	半濁点	半濁点	まる

すべての道はショートに通じる

入門ミニミニプログラム

Yoshida Kouichi
吉田 幸一

BASICに入門中の皆さん，どんなふうに勉強してますか？　命令ばかり暗記してもだめですよ。小さなプログラムをたくさん作ってみること，それがいちばん効率的で楽しい方法です。そんな基本テクニックを集めたショートプログラム集。一緒に楽しみましょう。

どんな巨大なプログラムも小さなプログラムの集合である。というわけで，ミニミニプログラムの作成に挑戦してみましょう。ここでは入門的なものからちょっと便利なプログラムまでを集めてみました。

「BASICはまだ始めたばかりでよくわかんないよー」という方。BASICを覚えるには，とにかくプログラムを作ってみること，プログラムを読んでそれを理解することがいちばんです。「テーマ」を読み，「リスト」を打ち込み，「解説」を見て勉強してください。

「BASICは今勉強中」という方。まずは「テーマ」だけを見て自分でプログラムを作ってみましょう。もしわからなかったら「解説」を読んでもかまいません。それでもできなかったら「リスト」を入力してみてください。掲載されている「リスト」はひとつのプログラム例ですから，皆さんの作ったものと違うと思います。プログラムの仕方・スタイルは人により千差万別ですから，違っていて当然なのです。要は目的どおりの動作をするかどうかです。

「BASICはだいたいマスターしたぜい」という方。力だめしに「テーマ」だけ読んでかたっぱしからプログラムを作ってみましょう。きっと作れないものがあるはずですよ。「リスト」と皆さんのプログラムが一致しないのは前述のとおりです。応用を見て，バージョンアップに挑戦してみてください。

それでは，Let's program！

テーマ1　四則演算

まず手始めに，2つの数AとBを入力すると，四則計算（A＋B，A－B，A＊B，A／B），およびAのB乗（A∧B）の計算を行い，答えを表示するプログラムを作りましょう。Bが0のときはA／Bでエラーが出るので，チェックして出ないようにします。

解説

INPUT文を2つ使ってAとBの変数に値を入力し，PRINT文で表示しています。ポイントはB＝0のときの処理で，ここではA／Bのみ“フノウ”と表示します。また，入力した数が大きすぎるとオーバーフローエラーが発生しますが，ここでは無視することにします。

このプログラムは無限ループになっていていつまでたっても終わらないので，止めたいときはブレークしてください。

応用

もう少し手を加えれば電卓代わりくらいには使えます。

テーマ2　三角関数

今度は，Aを入力するとその三角関数6種（sin, cos, tan, sec, cosec, cotan）を計算して表示するプログラムです。面倒なので，分母が0の場合やオーバーフロー時の処理はいらないものとします。

解説

入力した角度をラジアンに変換するために50行を設けてあります。また，内部演算での誤差のため，cos90°が0にならないな

リスト2　三角関数

```
 10 PRINT "*** ｻﾝｶｸ ｶﾝｽｳ ***"
 20 PRINT "ｶｸﾄﾞｦ ﾆｭｳﾘｮｸ ｼﾃ ｸﾀﾞｻｲ"
 30 PI=3.1415927
 40 INPUT "ｶｸﾄﾞ X= ";X
 50 X=PI*X/180
 60 PRINT "SIN(X)= ";SIN(X)
 70 PRINT "COS(X)= ";COS(X)
 80 PRINT "TAN(X)= ";TAN(X)
 90 PRINT "SEC(X)= ";1/COS(X)
100 PRINT "COSEC(X)= ";1/SIN(X)
110 PRINT "COTAN(X)= ";1/TAN(X)
120 END
```

リスト1　四則演算

```
 10 PRINT "***ｶｸｼｭ ｹｲｻﾝ***"
 20 PRINT "A ﾄ B ｦ ｲﾚﾃ ｸﾀﾞｻｲ"
 30 INPUT "A= ";A
 40 INPUT "B= ";B
 50 PRINT "A+B= ";A+B
 60 PRINT "A-B= ";A-B
 70 PRINT "A*B= ";A*B
 80 IF B=0 THEN PRINT "A/B= ﾌﾉｳ":GOTO 100
 90 PRINT "A/B= ";A/B
100 PRINT "A^B= ";A^B
110 GOTO 10
```

リスト3　数の和（1）

```
10 PRINT "*** ｺﾞｳｹｲ ｦ ﾀﾞｽ ***"
20 PRINT "0 ｦ ﾆｭｳﾘｮｸ ｽﾙﾄ ｺﾞｳｹｲｦ ｹｲｻﾝ ｼﾏｽ"
30 T=0
40 INPUT "A= (or 0)";A
50 IF A=0 THEN 80
60 T=T+A
70 GOTO 40
80 PRINT "ｺﾞｳｹｲ = ";T
90 END
```

どおかしな結果が出ます。

応用

誤差を小さくするために上何桁かで丸めたり，オーバーフロー時は“フノウ”と表示させたりすればもっと実用的になります。頑張ってください。

テーマ3　数の和(1)

数値をどんどん入力し，その和を出力するプログラムです。0を入力すると入力要求をやめ，和と計算結果を出力するようにしましょう。

解説

0を入力するまで入力要求をするのがポイントです。

合計を入れる変数Tを用意し，入力された数を次々と足していきます。0が入力されたか否かの判断は50行のIF～THENで行っています。これがないといつまでたっても結果は箱の中です。

応用

次の項を見てください。これの応用編になっています。

テーマ4　数の和(2)

前項と同じテーマですが，今回は結果を表示する際，一緒に計算式も表示させたいと思います。つまり，入力した数を保存しておきたいわけです。たとえば，最初に入力した数が5，次が3，そして0が入力されたとき，

5　＋　3　＝　8

と出力したいのです。配列をうまく使うのがポイントです。

解説

テーマ3と似ていますが，まったく違う構造になってます。大事なのは，変数の集まりである配列変数を40行で宣言し，50行で配列A(I)に数値を入力している点です。表示するときはFOR～NEXT文を使って，合計を計算しながら計算式を表示しています。ここが「FOR J＝0 TO I」でない点に注意してください。

応用

表計算をするにしろ，何にしろ，配列をうまく使わないと効率的なプログラミングをすることができません。さらに2次元配列を勉強して，表集計プログラムなどに挑戦してみてください。

テーマ5　英文作成

英単語をひとつずつ入力していきます。そして，ピリオドを入れると，今まで入力した英単語をつなげて英文にするプログラムを作りましょう。入力はinput文1回につき1単語を原則とします。

例）？ This

? is

? .

This is.

解説

文字列処理の基礎です。リストを見てわかるとおり，文字型配列に単語を入れます。表示するときはスペースを挿入しながら表示すればよいのですが，最後の単語のあとにスペースはいらないので，そこだけ別にピリオドを付けています。

応用

得られた文をひとつの文字列変数に入れると文字列のマージができます。マージというのは，平たく言えば2つ以上のものをくっつけることです。

テーマ6　名字と名前

前テーマと逆の作業です。氏名を入力すると，それを名字と名前に分けて出力するプログラムを作りましょう。入力時には名字と名前の間にひとつスペースを入れるようにします。INSTR関数はいちおう使わないでください。

例）？　ヨシダ　コウイチ

ミョウジ　：　ヨシダ

ナマエ　：　コウイチ

解説

ここでは文字列関数 MID$, RIGHT$, LENと，派手に使っています。MID$でスペースの位置を捜し，K$に名字を，残りをRIGHT$で取り出してJ$に名前というように使っています。

なお，テーマ1のように一見無限ループですが，“Z”を入力すると終了するようになっている点で進歩が見られます。

応用

スペースの数だけ分解できるようにすれば，英文を単語ごとに分けるといった芸当もできます。文字列処理の基本といえるでしょう。会話プログラムなどを作るときの重要なテクニックです。

テーマ7　統計

少し実用的なものに挑戦，ということで，各種統計計算をパソコンにさせてみましょ

リスト4　数の和(2)

```
 10 PRINT "*** ｺﾞｳｹｲ ｦ ﾀﾞｽ ***"
 20 PRINT "0 ｦ ﾆｭｳﾘｮｸ ｽﾙﾄ､"
 30 PRINT "      ｼｷ ﾄ ｹｯｶ ｦ ﾋｮｳｼﾞｼﾏｽ "
 40 DIM A(100):T=0:I=0
 50    INPUT "DATA= (or 0)";A(I)
 60    IF A(I)><0 THEN I=I+1:GOTO 50
 70 FOR J=0 TO I-2
 80    T=T+A(J):PRINT A(J);"+";
 90 NEXT J
100 T=T+A(I-1):PRINT A(I-1);"=";
110 PRINT T
120 END
```

リスト5　英文作成

```
 10 PRINT "*** ｴｲﾌﾞﾝｦ ﾂｸﾙ ***"
 20 PRINT " ｻｲｺﾞﾆ ﾋﾟﾘｵﾄﾞｦ ｲﾚﾃ ｸﾀﾞｻｲ"
 30 DIM E$(100)
 40 N=0
 50 INPUT "WORD = ";W$
 60 E$(N)=W$
 70 IF W$><"." THEN N=N+1:GOTO 50
 80 '
 90 PRINT
100 FOR I=0 TO N-2
110    PRINT E$(I);" ";
120 NEXT I
130 PRINT E$(N-1);"."
140 END
```

リスト6　名字と名前

```
 10 PRINT "*** ﾐｮｳｼﾞﾄ ﾅﾏｴﾆ ﾜｹﾙ ***"
 20 PRINT "ﾐｮｳｼﾞﾄ ﾅﾏｴﾉ ｱｲﾀﾞﾆ ｽﾍﾟｰｽｦ ｲﾚﾃｸﾀﾞｻｲ"
 30 K$="":N$="":J$=""
 40 INPUT "NAME= (or Z)";N$
 50 IF N$="Z" THEN END
 60 L=LEN(N$)
 70 FOR I=1 TO L
 80    M$=MID$(N$,I,1)
 90    IF M$><" " THEN K$=K$+M$ ELSE 120
100 NEXT I
110 GOTO 130
120 J$=RIGHT$(N$,L-I)
130 PRINT "ﾐｮｳｼﾞ    : ";K$
140 PRINT "ﾅﾏｴ      : ";J$
150 GOTO 30
```

う。入力形式はテーマ4と同じにします。

各種といっても統計計算にはいろいろあるので，ここでは以下の3つについて行ってみてください。

1）平均

2）分散 $(\sum_{I=1}^{N} A(I)^2) \div N - M^2$

つまり，(入力データを2乗したものの和)÷(データ数)－(平均)^2

3）標準偏差 $\sqrt{(分散)}$

解説

テーマ4の要領で次々とデータを入力していき，0を入れると計算を始めます。

統計計算についての詳しくは専門の本でも見てください。大事なのは計算式ではなく，前の6つよりグンと実用的になった点です。

応用

ついでですから，不偏分散を求めたり度数分布図を描いたり，調和平均や変動係数が求まるようにしてみてはいかがでしょう。

テーマ8　三角形の面積

いく通りかの方法で三角形の面積を求めてみましょう。というだけでは抽象的なので，底辺と高さ，3辺の長さ，2辺とその間の角のどれかがわかっていれば面積が得られるようにします。

入力は，まず3つのどの方法で求めるかを選んでからデータを入れる，としたほうが楽ですね。ちなみに，計算式は下の2つでこと足ります。

1）底辺×高さ÷2

2）ヘロンの公式

面積$=\sqrt{s(s-a)(s-b)(s-c)}$

(a, b, cは辺の長さ, $s=\frac{(a+b+c)}{2}$)

2辺とその間の角から求めるには，三角関数を使います。

解説

底辺と高さを入力したときについては言うまでもないですね。3辺の長さから求めるにはヘロンの公式

面積$=\sqrt{s(s-a)(s-b)(s-c)}$

$s=(a+b+c)/2$

を使います。2辺とその間の角から求めるには，三角関数を使って底辺と高さの式で求まるように変換しています。求め方の選択は，ON～GOSUB文を使って適当なサブルーチンを呼んでいます。

応用

これをさらに発展させ，平行四辺形や円の面積も求まるようにしたらおもしろいのではないでしょうか。

テーマ9 ソート(並べ換え)

いくつかの数を入力すると，小さい順(昇順)に並べ換えて表示するプログラムを作りましょう。いわゆるソートです。

入力はテーマ4と同じにします。ソートの方法はいろいろありますが，ここでは簡単な「隣の数と比べて逆にする必要があれば2つをひっくり返す」ことを繰り返す方法をとります。配列の初めの方から順に小さい数が入っていくようにしてみてください。

解説

配列の中からいちばん小さい数を見つけていちばん前に，2番めに小さい数を2番めに…… としてソートしています。あとはN(0)から順に表示するだけです。80行の“<=”を“>=”にすると大きい順（降順）にソートします。

応用

10個や20個のソートならこれでもいいのですが，100個や1000個のデータをソートするとなるともっと早い方法はないのか！と言いたくなります。というわけで，次の項でもう少し速いソートを紹介します。

テーマ10　シェルソート

テーマ9の内容をシェルソートという方法で行ってみましょう。

シェルソートというのは有名なわりにいまひとつ説明しづらいので，知らない人はプログラムをカンニングしてください。作られたプログラムの解読もひとつの勉強です。

解説

シェルソートは別名跳び跳びソートとも言われ，比較的速いソートとして有名です。適当な数を入れてトレースしてみる（流れを追ってみる）とわかりますが，なぜかこれでソートができてしまうのです。プログラムの途中で，変数STやL，配列の内容などを表示させると流れを追いやすいと思います。

応用

住所録など大量のデータ処理を行うとき，五十音順や電話番号順などのソートが必要となります。こういったときにシェルソートは威力を発揮するのです。サブルーチンとしてのほかのプログラムに応用してみてください。

リスト7　統計計算

```
 10 PRINT "*** ﾄｳｹｲ ｹｲｻﾝ ***"
 20 PRINT "0 ｦ ﾆｭｳﾘｮｸ ｽﾙﾄ､"
 30 PRINT "            ｹｯｶ ｦ ﾋｮｳｼﾞｼﾏｽ "
 40 DIM A(100):T=0:T2=0:I=1
 50    INPUT "DATA= (or 0)";A(I)
 60    IF A(I)><0 THEN I=I+1:GOTO 50
 70    N=I-1
 80 FOR J=1 TO N
 90    T=T+A(J)
100    T2=T2+A(J)*A(J)
110 NEXT J
120    M=T/N
125    S2=(T2-T*T/N)/N
126    S=SQR(S2)
130 PRINT "********************************"
140 PRINT "      ｿｳ ｺｽｳ=";N
145 PRINT "        ｿｳ ﾜ=";T
146 PRINT "        ﾍｲｷﾝ=";M
148 PRINT "       ﾌﾞﾝｻﾝ=";S2
149 PRINT "ﾋｮｳｼﾞｭﾝ ﾍﾝｻ=";S
150 PRINT "********************************"
160 END
```

リスト8　三角形の面積

```
 10 PRINT "*** ｻﾝｶｸｹｲ ﾉ ﾒﾝｾｷ ***"
 20 PRINT " 1:ﾃｲﾍﾝﾄ ﾀｶｻｶﾞ ﾜｶｯﾃｲﾙ"
 30 PRINT " 2:3ﾍﾝﾉ ﾅｶﾞｻｶﾞ ﾜｶｯﾃｲﾙ"
 40 PRINT " 3:2ﾍﾝﾉ ﾅｶﾞｻﾄ ｿﾉ ｱｲﾀﾞﾉｶｸｶﾞ ﾜｶｯﾃｲﾙ"
 50 INPUT "(1-3)";N:IF 0>N OR N>3 THEN 50
 60 ON N GOSUB 90,130,190
 70 PRINT "ﾒﾝｾｷ = ";S
 80 GOTO 10
 90 ' No.1
100 INPUT "ﾃｲﾍﾝ = ";L
110 INPUT "ﾀｶｻ  = ";H
120 S=L*H/2:RETURN
130 ' No.2
140 FOR I=1 TO 3
150    PRINT "ﾍﾝ";I;"= ";:INPUT HL(I):P=P+HL(I)
160 NEXT I
170 P=P/2:S=P*(P-HL(1))*(P-HL(2))*(P-HL(3))
180 S=SQR(S):RETURN
190 ' No.3
200 INPUT "ﾍﾝ 1 = ";L1
210 INPUT "ﾍﾝ 2 = ";L2
220 INPUT "ｶｸﾄﾞ = ";R
230 R=3.1415972/180*R:H=L2*SIN(R)
240 S=L1*H/2:RETURN
```

テーマ11　俳句を詠む

日本語処理の第一歩(?)として，パソコンに俳句を詠ませてみましょう。5,7,5であれば文字は乱数発生でよい。

ここでは，あらかじめDATA文にア～ンの45文字を入れておき，それをREAD文で配列に読んで，ランダムに出力するといった方法をとります。

解説

今回はデータ文にア～ンを入れて使用しました。READ文で勝手にDATA文中のデータを順に読んでくれるので，あとはFOR～NEXTをデータの数だけ回し配列に突っ込んでやれば，あとはいつでも呼び出せます。そして，RND関数とINT関数を使い0～44の整数をランダムに発生させ，それに対応する文字を表示しています。これをFOR～NEXTを使って5文字，スペース，7文字，スペース，5文字と書けば俳句の完成です。

応用

簡単な応用としては，DATA文中に濁音，半濁音(ガやパ)を入れる，READ～DATAを使わずにCHR$関数を使う，などがあげられます。少し高度なものとしては，日本語の文章を研究して使用頻度の高さを考慮して文字を発生させれば多少日本語らしい文が作れるかもしれませんね。

テーマ12　10進→n進変換

10進法で入力した数を任意の進数（3進や15進）に変換して出力するプログラムです。当然、11進数以上に変換するときはアルファベットも使うことにします。

まず何進に変換したいかを入力したあと，変換もとの10進法を入れるようにしましょう。n進数の意味を知らない人は……　調べてください。

解説

こういったプログラムのアルゴリズムはたいへん重要なので，しっかり理解してください。たとえば，10進の38は38を10で割った余り（つまり8）とその商を10で割った余り（つまり3）を右から順に書いたものと考えます。この調子で38を8進にしてみると，38を8で割った余り6とその商4を8で割った余りで46，となるわけです。この作業を80～150行で行っています。

このプログラムのもうひとつのポイントはCHR$を使っている点です。CHR$(&H30）は数字の“0”，CHR$(&H41)は“A”で，このほうが11進以上でアルファベットを使いたいときに便利なうえ，文字列として処理すると前に書いた「右から順に書く」ときに都合がいいのです。

応用

もっと実用性を上げるには，n進↔n進変換ができるように機能アップするなり，2進↔10進↔16進変換表を作ってプリントアウトしてはいかがでしょう。

テーマ13　ミニミニ電話帳

名前から電話番号を検索するミニミニ電話帳を作りましょう。名前と電話番号はあらかじめDATA文中に入れておくことにします。

名前を入力すると，その人のフルネームと電話番号を表示するようにします。名前の入力は名字でも頭文字だけでもよいようにし，その場合は該当する人すべてを表示します。

リスト9　ソート(並べ換え)

```
 10 PRINT "*** ｶｽﾞﾉ ｿｰﾄ ***"
 20 DIM A(100):I=0
 30 INPUT "N= (or 0)";A(I)
 40 IF A(I)><0 THEN I=I+1:GOTO 30
 50 '
 60 FOR J=0 TO I-2
 70   FOR K=J TO 0 STEP -1
 80      IF A(K)<=A(K+1) THEN 100
 85      B=A(K):A(K)=A(K+1):A(K+1)=B
 90   NEXT K
100 NEXT J
110 '
120 FOR A=0 TO I-1
130   PRINT A(A)
140 NEXT A
150 END
```

リスト11　俳句を詠む

```
 10 PRINT "*** ﾊｲｸ ｦ ﾖﾑ ***"
 20 DIM N$(44)
 30 FOR I=0 TO 44
 40    READ I$:N$(I)=I$
 50 NEXT I
 60 FOR I=1 TO 5
 70    N=INT(RND(1)*45):PRINT N$(N);
 80 NEXT I:PRINT " ";
 90 FOR I=1 TO 7
100    N=INT(RND(1)*45):PRINT N$(N);
110 NEXT I:PRINT " ";
120 FOR I=1 TO 5
130    N=INT(RND(1)*45):PRINT N$(N);
140 NEXT I:PRINT " ";
150 END
160 DATA ｱ,ｲ,ｳ,ｴ,ｵ,ｶ,ｷ,ｸ,ｹ,ｺ,ｻ,ｼ,ｽ,ｾ,ｿ
170 DATA ﾀ,ﾁ,ﾂ,ﾃ,ﾄ,ﾅ,ﾆ,ﾇ,ﾈ,ﾉ,ﾊ,ﾋ,ﾌ,ﾍ,ﾎ
180 DATA ﾏ,ﾐ,ﾑ,ﾒ,ﾓ,ﾔ,ﾕ,ﾖ,ﾗ,ﾘ,ﾙ,ﾚ,ﾛ,ﾜ,ﾝ
```

リスト10　シェルシート

```
 10 PRINT "*** ｼｪﾙ ｿｰﾄ ***"
 20 DIM N(100):C=0
 30 INPUT "N= (or 0)";N(C)
 40 IF N(C)><0 THEN C=C+1:GOTO 30
 50 '
 60 C=C-1:ST=C
 70 ST=INT(ST/2)
 80 FOR L=0 TO C-ST
 90   FOR U=L TO 0 STEP -ST
100      IF N(U)<=N(U+ST) THEN 130
110      A=N(U):N(U)=N(U+ST):N(U+ST)=A
120   NEXT U
130 NEXT L
140 IF ST>1 THEN 70
150 '
160 FOR A=0 TO C
170   PRINT N(A)
180 NEXT A
190 END
```

リスト12　10進→n進変換

```
 10 DIM C$(30)
 20 I=0
 30 INPUT "n=",N
 40 IF N>36 THEN 30
 50 INPUT "10ｼﾝｽｳ=",D
 60 PRINT:PRINT "10 ｼﾝｽｳ ﾉ ｶｽﾞ ";D;"ｦ ";N;"ｼﾝｽｳ ﾆ ﾍﾝｶﾝ ｼﾏｽ "
 70 PRINT:PRINT N;"ｼﾝｽｳ=";
 80 K=D-INT(D/N)*N
 90 IF D=0 THEN 130
100 IF (0<=K)*(K<=9) THEN C$(I)=CHR$(48+K):GOTO 120
110 C$(I)=CHR$(64+K-9)
120 D=INT(D/N):I=I+1:GOTO 80
130 FOR J=I TO 0 STEP -1
140    PRINT C$(J);
150 NEXT
160 END
```

解説

最初のFOR～NEXTブロックで名前を全部配列にしまい，Zを読むと次のFOR～NEXTブロックへ行き，電話番号を配列に取り込みます。つまり，N$(0, I)の人の電話番号はN$(1, I)なわけです。

次に検索ですが，ここでは入力した文字列と同じ文字列で始まる名前を捜すようになっています。たとえば“イ”を入力すると，イトウ～イチカワの2人が，“イト”と入力するとイトウが検索されるのです。

応用

簡単なところでは，検索した結果をプリントアウトして使ったり，年齢や住所などのデータも入れて汎用性を上げたりすればもう少し実用的になりますね。

テーマ14 ミニ電話帳

テーマ13を拡張し，シーケンシャルファイルを使ったミニ電話帳を作ってみましょう。データは前と同じく名前と電話番号だけにし，メニューは

1．DATAニュウリョク（ツイカ）
2．DATAサンショウ
3．DATA SAVE
4．DATA LOAD
5．END

とします。

解説

今までのものと比べるとかなり長いプログラムですが，その分丁寧に解説しますのでご了承を。

初めて使うときは“DATA”という名前のファイルをひとつ作っておいてください。そうしないと270行のKILL文でエラーが出ます。

さて，DATA入力モードです。ここでは，名前と電話番号を配列に読み込んでいます。あらかじめデータがロードされているときは，Iに最後に入力したデータの番号が入っているので，このルーチンでデータ追加ができます。

DATA参照モードは前項と同一なので省略するとして，DATAセーブモードへと話は移ります。ここでは，元のデータファイルを消してから，書き込みモードでファイルオープンし，配列の内容をすべてPRINT＃1でディスクあるいはテープに書き込んでいます。最後に“＊＊＊”を書いているのは，データロード時にデータ終了を知らせるためです。

かくして，DATAロードモードです。ここでは読み込みモードでファイルをオープンし，“＊＊＊”が見つかるまで配列に読み込んでいます。この際，Iに最後のデータ

リスト14 ミニ電話帳

```
 10 PRINT "*** ﾐﾆ ｱﾄﾞﾚｽ ﾌﾞｯｸ ***"
 20 DIM NT$(1,100)
 30 PRINT "1:DATA ﾆｭｳﾘｮｸ(ﾂｲｶ)"
 40 PRINT "2:DATA ｹﾝｻｸ"
 50 PRINT "3:DATA SAVE"
 60 PRINT "4:DATA LOAD"
 70 PRINT "5:END"
 80 INPUT "(1-5)";M:IF (M<1)+(M>5) THEN 80
 90 ON M GOSUB 110,170,260,350,420
100 GOTO 30
110 'DATA ﾆｭｳﾘｮｸ(ﾂｲｶ)
120 INPUT "NAME (or Z): ";N$:IF N$="Z" THEN 160
130 INPUT "TEL        : ";T$
140 I=I+1:NT$(0,I)=N$:NT$(1,I)=T$
150 GOTO 120
160 RETURN
170 'DATA ｻﾝｼｮｳ
180 INPUT "NAME(or Z) : ";N$:IF N$="Z" THEN RETURN
190 L=LEN(N$)
200 FOR J=1 TO I
210    IF L>LEN(NT$(0,J)) THEN 240
220    IF LEFT$(NT$(0,J),L)><N$ THEN 240
230    PRINT "ﾅﾏｴ  : ";NT$(0,J):PRINT "TEL  : ";NT$(1,J)
240 NEXT J
250 GOTO 180
260 ' DATA SAVE
270 KILL "DATA"
280 OPEN "O",#1,"DATA"
290 FOR J=1 TO I
300    PRINT #1,NT$(0,J);",";NT$(1,J)
310 NEXT J
320 PRINT #1,"***,***"
330 PRINT "COMPLETED"
340 CLOSE #1:RETURN
350 ' DATA LOAD
360 OPEN "I",#1,"DATA"
370 FOR J=1 TO 100
380    INPUT #1,NT$(0,J),NT$(1,J):IF NT$(0,J)="***" THEN I=J-1:J=100
390 NEXT J
400 PRINT "COMPLETED"
410 CLOSE #1:RETURN
420 END
```

●S-BASIC用変更点

```
270 DELETE "DATA"
280 WOPEN# 1,"DATA"
360 ROPEN# 1,"DATA"
```

リスト13 ミニミニ電話帳

```
 10 PRINT "*** ﾐﾆﾐﾆ ｱﾄﾞﾚｽ ﾌﾞｯｸ ***"
 20 DIM N$(1,100)
 30 FOR I=0 TO 100
 40    READ NA$:IF NA$="Z" THEN 70
 50    N$(0,I)=NA$
 60 NEXT I
 70 FOR A=0 TO I-1
 80    READ NA$:N$(1,A)=NA$
 90 NEXT A
100 '
110 PRINT "ﾅﾏｴ ｶﾗ TEL No. ｦ ｹﾝｻｸ ｼﾏｽ "
120 PRINT " Z ｦ ｲﾚﾙﾄ ｵﾜﾘﾏｽ"
130 INPUT "NAME = ";NA$
140 IF NA$="Z" THEN END
150 NL=LEN(NA$)
160 FOR A=0 TO I-1
170    IF NL>LEN(N$(0,A)) THEN 200
180    IF LEFT$(N$(0,A),NL)><NA$ THEN 200
190    PRINT "ﾅﾏｴ : ";N$(0,A):PRINT "TEL : ";N$(1,A)
200 NEXT A
210 GOTO 130
220 DATA ﾖｼﾀﾞ ｺｳｲﾁ,ｲｽﾞﾐ ﾀﾞｲｽｹ,ｻﾄｳ ｱｷﾗ,ｲﾜｲ ｲｯﾍﾟｲ,ｼﾐｽﾞ ｶｽﾞﾄ,ｺｳﾓﾄ ﾔｽﾋｺ,ｻﾄｳ ﾜｶｺ,Z
230 DATA 0423-85-0000,03-383-0000,0486-25-0000,03-265-0000,06-583-0000,052-771
-0000,0422-47-0000
```

が入った配列の番号を代入しています。

このように外部記憶装置にデータを保存するとメリットがたくさん出てきます。データの追加の楽なことや，何本もメディア(テープやディスク)を用意すれば，それぞれ違った種類のデータに使えることです。

応用

今回は100件までしかデータが入りませんが（20行）もっとたくさん入るようにしたり，データの訂正やプリントアウトができるようにするともっと実用的になるでしょう。

また，フロッピーディスクをお持ちの方はランダムアクセスファイルに挑戦してみてください。

テーマ15　四角形を描く

ちょっと固いテーマが多くなってしまいました。ここらで柔らかい話題にテーマを移しましょう。

任意のキャラクタで，一辺の長さが任意の正方形を描くプログラムです。

たとえば，1辺の長さが5でキャラクタが@と入力すると，1辺に@が5個並んだ正方形が表示されるようにしてください。

解説

これは，グラフィック機能を使わずに図形を描くときの基本です。プログラムは簡単なのでたいして説明の必要はないでしょう。FOR～NEXT中で，入力したキャラクタをその数に応じて表示しているだけです。PRINT文中にあるSPCは引数の数だけスペースを書くための命令でPRINT文中でしか使えません。

応用

三角形や大きな文字を書くことも当然できます。ゲームのタイトルなど，派手な字を書きたいときに，キャラクタを並べて字を書く方法はたいへんよく使われます。

テーマ16　お絵描き

カーソルキーを使って，任意のキャラクタで絵を描くプログラムを作りましょう。カーソル位置は常に画面の左上に表示するようにします。

カーソルキーを押すと，その方向に画面に出ているキャラクタを表示します。また，Aキーを押すとAが表示されるようにします。たとえば，Aを押して「→」キーを押すと，その位置からAAA……とカーソルを離すまで描き続けるわけです。

解説

言うまでもなく，前テーマの応用編です。70行でゲームによく使うINKEY$関数を使っていますが，この関数にはそのときに押されていたキーが入っています。何も押してなければ“”が入ります。そして80～110行でカーソル位置を計算し，120行で表示する文字を決めているわけです。グラフィックエディタの原生動物みたいなものですね。

応用

とにかくゲームに使えます。自分の戦闘機を動かしたり，敵機を動かしたりはたいていこれが基本です。

テーマ17　パソコンオルガン

キーを押すと音が鳴る，つまりパソコンをオルガンのように使ってみましょう。出力は1オクターブのみで，キーボード上の「C」をドとします。

解説

Cをドとし1オクターブ上のドまで音が出るようになっています。アルゴリズムはいたって簡単。どのキーを押したかの判定は，使う13個のキーをあらかじめ KB＄に入れておき(10行)，80～100行で，押され

リスト16　お絵描き

```
 10 CLS
 20 X=20:Y=10
 30 B$="+"
 40 LOCATE 0,0:PRINT "                              ";
 50 LOCATE 0,0:PRINT "X=";X;"  Y=";Y
 60 LOCATE X,Y:PRINT B$;
 70 A$=INKEY$:IF A$="" THEN 70
 80 IF A$=CHR$(30) AND Y>1 THEN Y=Y-1
 90 IF A$=CHR$(31) AND Y<18 THEN Y=Y+1
100 IF A$=CHR$(29) AND X>0 THEN X=X-1
110 IF A$=CHR$(28) AND X<38 THEN X=X+1
120 IF ASC(A$)>31 AND ASC(A$)<128 THEN B$=A$
130 GOTO 40
```

●S-BASIC用変更点

```
 10 PRINT CHR$(6)                 :' K/C ｼﾘｰｽﾞﾊ CHR$(22)
 40 CURSOR 0,0:PRINT "                              ";
 50 CURSOR 0,0:PRINT "X=";X;"  Y=";Y
 60 CURSOR X,Y:PRINT B$;
 70 GET A$   :IF A$="" THEN 70
 80 IF (A$=CHR$(2))*(Y>1) THEN Y=Y-1            :' K/C ｼﾘｰｽﾞﾊ CHR$(18)
 90 IF (A$=CHR$(1))*(Y<18) THEN Y=Y+1           :' K/C ｼﾘｰｽﾞﾊ CHR$(17)
100 IF (A$=CHR$(4))*(X>0) THEN X=X-1            :' K/C ｼﾘｰｽﾞﾊ CHR$(20)
110 IF (A$=CHR$(3))*(X<38) THEN X=X+1           :' K/C ｼﾘｰｽﾞﾊ CHR$(19)
120 IF (ASC(A$)>31)*(ASC(A$)<128) THEN B$=A$
```

リスト15　四角形を描く

```
 10 PRINT "*** 4ｶｸｹｲ ***"
 20 INPUT "1ﾍﾝﾉ ﾅｶﾞｻ    = ";N
 30 IF N>20 THEN 20
 35 INPUT "ｷｬﾗｸﾀｰ(1ﾓｼﾞ)= ";N$
 36 IF LEN(N$)>1 THEN 35
 40 FOR I=1 TO N
 50    PRINT N$;
 60 NEXT I
 70 PRINT
 80 FOR I=1 TO N-2
 90    PRINT N$;SPC(N-2);N$
100 NEXT I
110 FOR I=1 TO N
120    PRINT N$;
130 NEXT I
140 END
```

リスト17　パソコンオルガン

```
 10 DIM P$(13):KB$="CVBNM,./FGJKL"
 20 PRINT "*** ﾊﾟｿｺﾝ ｵﾙｶﾞﾝ ***"
 30 FOR A=1 TO 13
 40  READ A$:P$(A)=A$
 50 NEXT A
 70 I$="":I$=INKEY$:IF I$="" THEN 70
 80 FOR A=1 TO 13
 90    IF I$=MID$(KB$,A,1) THEN 110
100 NEXT A:GOTO 70
110 PLAY P$(A)
120 GOTO 70
130 DATA O4C8,O4D8,O4E8,O4F8,O4G8,O4A8,O4B8,O5C8,O4C+8,O4D+8,O4F+8,O4G+8,O4A+8
```

●S-BASIC用変更点

```
 70 I$="":GET I$   :IF I$="" THEN 70
110 MUSIC P$(A)
130 DATA C3,D3,E3,F3,G3,A3,B3,+C3,#C3,#D3,#F3,#G3,#A3
```

たキーがKB＄の中にあったかどうか捜すことによってなされています。配列P＄に鳴らす音のデータが入れてあるので，KB$の5番目(“M”)が押されたらP$(5)(“O4G8”つまりソの音）を鳴らしてやればよいのです。

さまざまな欠点をかかえつつも，けっこう遊べるプログラムになっています。

応用

パソコンをシンセサイザのように使う試みは昔からなされてきました。BASICでもかなりのことができますが，本格的に使おうと思うとマシン語の世界に足を踏み入れざるをえません。どうしましょう。

テーマ18　サイコロ

サイコロのシミュレーションをしてみましょう。できれば出た目のビジュアルな表現をしたいですね。

スペースを押すとサイコロが振られるようにします。ビジュアルな表現といっても，テーマ15のようにキャラクタを並べて絵を描く程度でいいでしょう。

解説

これは乱数発生とテーマ15のグラフィック表現に加え，INKEY$とON～GOSUBを組み合わせたものです。

見てのとおり，スペースキーを押すとサイコロが振られ，画面に出た目が表示されます。絵はサブルーチンで描いており，乱数で出た値によって，ON～GOSUB文でその目を描くサブルーチンを呼んでいるのです。

応用

スゴロクに使うよりも，今はやりのアドベンチャーゲームブック用に使えそうです。ゲームに応じて，一度に2つサイコロを振ったり，戦闘力を計算したりするのはいかがでしょう。

テーマ19　早押しゲーム

最後はキーボードの早押しゲームです。ルールは簡単。AからZまでを順に押し，押し終えるまでにかかった時間を表示するものです。「A」を押したらタイマーがスタートするようにして正確に判定しましょう。

解説

「A」を押した時点でタイマーがスタートし，Zを押した時点でかかった時間を表示します。筆者のベストタイムは今のところ5秒です。5秒の壁は厚い。ちなみに，7秒でできれば速いほうでしょう。単純なわりに熱中できるのがこれの特徴です。

プログラムですが，難しいことはしてません。CHR$(&H41)が“A”,CHR$(&H5A)が“Z”だと覚えておけばすぐにわかるでしょう。

応用

打つ英字を英単語にしたりBASICの命令にすれば，タイピング練習プログラムとしても使えます。

以上，ミニミニプログラム集はいかがだったでしょうか。日頃気付かなかった新しい発見もあったのではないでしょうか。今は必要ないと思ったプログラムでも，あとで意外なところで役に立つことがあるかもしれません。

どんなに大きなプログラムも，小さなプログラム，小さなルーチンから成り立っています。また，BASICは小さなプログラムをちょこちょこといじって遊ぶのには最適な道具です。BASICを勉強する，BASICで遊ぶ，大きなプログラムを作る，すべてはショートプログラムから始まります。

リスト18　サイコロ

```
10 PRINT "--- ｻｲｺﾛ ｼﾐｭﾚｰｼｮﾝ ---"
20 PRINT "HIT SPACE BAR"
30 I$=INKEY$:IF I$><" " THEN 30
40 S=INT(RND(1)*6)+1
50 PRINT "***********":PRINT "*         *"
60 ON S GOSUB 90,130,170,210,250,290
70 PRINT "*         *":PRINT "***********"
80 GOTO 10
90 ' 1
100 PRINT "*         *":PRINT "*   ●    *"
110 PRINT "*  ●●●  *":PRINT "*   ●    *"
120 PRINT "*         *":RETURN
130 ' 2
140 PRINT "* ●      *":PRINT "*         *"
150 PRINT "*         *":PRINT "*         *"
160 PRINT "*      ● *":RETURN
170 ' 3
180 PRINT "* ●      *":PRINT "*         *"
190 PRINT "*   ●    *":PRINT "*         *"
200 PRINT "*      ● *":RETURN
210 ' 4
220 PRINT "* ●   ● *":PRINT "*         *"
230 PRINT "*         *":PRINT "*         *"
240 PRINT "* ●   ● *":RETURN
250 ' 5
260 PRINT "* ●   ● *":PRINT "*         *"
270 PRINT "*   ●    *":PRINT "*         *"
280 PRINT "* ●   ● *":RETURN
290 ' 6
300 PRINT "* ●   ● *":PRINT "*         *"
310 PRINT "* ●   ● *":PRINT "*         *"
320 PRINT "* ●   ● *":RETURN
```

●S-BASIC用変更点

```
30 GET I$:IF I$><" " THEN 30
```

リスト19　早押しゲーム

```
10 PRINT "*** ﾊﾔｵｼ ｹﾞｰﾑ ***"
20 PRINT " A-Z ﾉ ｷｰｦ ｼﾞｭﾝ ﾆ ｵｼﾃ ｸﾀﾞｻｲ"
30 PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
40 IF INKEY$<>CHR$(65) THEN 40
50 TIME$="00:00:00"
60 PRINT "A";
70 FOR I=66 TO 90
80 A$=INKEY$
90 IF A$<>CHR$(I) THEN 80
100 PRINT CHR$(I);
110 NEXT
120 PRINT:PRINT TIME$
130 END
```

●S-BASIC用変更点

```
40 GET I$:IF I$<>CHR$(65) THEN 40
50 TI$="00:00:00"
80 GET A$
120 PRINT:PRINT TI$
```

雑誌のプログラムを完動させる

必殺バグ退治のテクニック

Asano Keizo
浅野 恵造

雑誌に掲載されているプログラムを入力して正しく動作させる。簡単そうでいて，じつは難しいことですね。入力ミスさえしなければよいのですが，やっぱり打ち間違ってしまいます。それらを効率的に直すためのデバッグのテクニックを勉強しましょう。

Oh!MZの読者（別にOh!MZに限りませんが）の皆さんは，多かれ少なかれ雑誌に掲載されているプログラムリストを自分のパソコンに入力された経験がおありだと思います。さて，そこで質問ですが，そのプログラムは一度で正しく動いたでしょうか？まあ多くの方の返事は「NO」だと思います。中には「YES」と自信を持って答えることのできる方もいるでしょうが，そうでない多くの方はこの講座をぜひ参考にしてください。

さて，次なる質問です。「NO」と答えられた方は最終的にそのプログラムを正常にできたでしょうか？これに対しては「YES」という方も多いと思いますが，中には「no」と小文字で小さく答えられた方もいらっしゃると思います。そんな方も，この記事を読んでいただければもう安心。これからは今まで諦めていたような長いリストも無事に入力できるようになります。

どうして正常に動かないのか

入力したプログラムが正常に動かない原因としてはいろいろ考えられます。操作ミス，入力ミス，プログラム自身の間違いや機械のトラブルなどです。操作ミスというのは，違うキーを押したりしてしまうことによるもので，CAPSキーやカナキーがロックされていたり，数字と文字の間違いなどによることが多いようです。これは，プログラムについての説明をよく読めば解決することであり，プログラム自身にそうした誤操作への配慮が払われるべきことなので別の機会に譲ります。また，プログラム自身の間違いというのもひとつの大きな問題ですがここでは触れません。したがって，雑誌に掲載されているリストは完璧である（この点について異議を唱える方も若干いることと思いますが私は関知しない）という前提のもと，入力ミスをいかに発見しどう修正していくかについての方法を述べたいと思います。というのは，きちんとした入力チェックをする前からプログラム自身を疑ってしまっては，永遠に完動させることはできないからです。

リストの正しい入力法

第1段階としては，入力をいかに正しく行うかということです。話の本筋は入力ミスの修正ですが，早い話が入力ミスがなければそんな心配はいらないわけですから，まずできるだけ入力を正確に行うための工夫を考えてみます。

1) 入力を共同で行う

これは私も実践していますが，ひとりにリストを読み上げてもらい，もうひとりがキーボードを叩くという方法です。ブラインドタイプができる人はこんなことをする必要はありませんが，キーボード操作に慣れていない場合（キーボードを見ながら打つ人）には特に有効です。リスト，キーボードそしてディスプレイと視線を移動させるとどうしても文字の見間違いや見落としがあるものです。この場合のパートナーはBASICなどパソコンについて知っている人のほうが楽でしょうが，とりあえずアルファベットが読める人であればかまいません。できればコマンドや記号がきちんと読めるほうがいいので，『音読講座』で勉強することをおすすめします。

2) 何回かに分けて入力する

これは，ある程度長いリストになると途中で疲れてしまい必然的にそういう結果になってしまうことです。しかし，意識的に分割して入力していくと間違いが減ると思います。区切りは行番号によってでもルーチンごとでもかまいません。人間はパソコンと違い長時間作業をすると疲れてミスも増えるので，作業を細かくすることで集中力がアップします。また，分割した区切りごとにそのつどセーブしておけば，停電，暴走その他の不測の事態にも対応できるなどの効果があります。

以上のほかにも人により自分に合った入力法があるでしょう。要は入力という単調作業に伴うミスを少しでも減らすように工夫することです。

入力のとき打ち間違いやすい文字は，形の似ているものとして1とIとl，：（コロン）と；（セミコロン），．（ピリオド）と，（カンマ），0（ゼロ）とO（オー）などがありますので特に注意してください。それから，キーボード上で隣り合っている文字，MとN，VとB，RとT，YとUなども比較的打ち間違いが多いようです。

さて，上のような間違い（勘違い）は誰でも起こし得るわけですが，これとは別に多少BASICなどの知識がある人にある入力ミスとして思い込みや勝手な解釈によるものがあります。これは入力の際に多少の知識があるばかりにリストで間違いと思われる部分，あるいは省略可能と思われる部分

を自分の判断で行ってしまうために起こるものです。たとえば，DATA文の，（カンマ）の省略，変数名の短縮，パラメータの省略，スペースの省略，REM文やPRINT文中の変更や省略などといったものです。こうしたことは慎重に行わないと，一見して正しいようでもプログラム全体で違う動作をしてしまうなどという，発見しづらい間違いとなる可能性があります。自分では正しいと思い込んでいるだけに大変です。ですから，最初はリストどおり（一字一句違わず）入力することをおすすめします。正常な動作を確認した上で自分に合うように改造すればよいのですから。

ただし，BASICの知識があるに越したことはありません。コマンドやパラメータの入力ミスは大幅に少なくなります。ただ，その知識をむやみに振り回さないことが大切なのです。

走らない！ さあどうするか

RUNする前のセーブはもう習慣になっていますか？　というわけで，今までのは序論として，いくら必死に入力しても間違いはあるものですから，これからは間違い(バグ：bug)を直す(デバッグ：debug)実際のアプローチについて見てみましょう。

バグがある場合のプログラムの動作としては2通りあります。ひとつはビープ音(「ブー」，「ピー」，「ポッ」etc.)と同時にプログラムの実行が止まり，エラーメッセージが表示される場合，もうひとつはエラーは出ないものの動作が異常（暴走も含むがBASICプログラムでの暴走は少ない）な場合です。後者のデバッグは多少難しいのであとで説明することにして，まずエラーメッセージが表示された場合のデバッグを考えます。

エラーメッセージというのは，コンピュータが実行できないような命令を与えられた場合，どういった種類の間違いかをユーザーに知らせるメッセージで，「Syntax error in 100」のようなものです。これがBASICインタプリタが対話型の言語であるという点でもありますが，ビープ音とともに予告なく現れるこのメッセージは，私にはケンカを売られているとしか思えず，とても対話などという気分はしません。それはともかく，売られたケンカは買わなくてはいけないので正しいケンカの仕方を考えてみましょう。

答えはエラーメッセージにあり

エラーメッセージの表示のされ方は，前の例にもあるとおり，「○○○○○in×××」といった形式です。in×××の×××はエラーの発生した行番号（バグのある行番号ではないことに注意)を示します。したがって，デバッグの第一歩はエラーの発生した行を調べることにあります。このときは全体のリストを表示させず，その行だけのリストを表示させます。L..⏎やE..⏎とするとストップした行のリストを表示するBASICもあるのでうまく活用しましょう（どちらも2つ目のピリオドはエラーの発生した行番号，すなわち×××の値の代わりをします)。そこで，雑誌のリストと比較して間違っていたら正しく打ち直します。入力法で触れた似た文字やキーボード上の近い文字といった打ち間違いが多いと思います。ここで1行だけ表示させると言ったのは，入力の際に2行分（行番号で2つ分ということ）が連結されてしまっている場合がたまにあるからで，1行だけ表示させたつもりが2行分のようなリストになって表示された場合がそれです。

さて，一見して間違っていないようなのにエラーが発生してしまう場合があります。これはほかの行に原因がある場合もあり，それはすぐあとで触れるとして，原因がその行にあるのに発見できないことがあります。そこで，ひとつの有効な手段として同じ行をもう一度打ち直してみるというのがあります。眺めただけでは発見しにくい打ち間違いも再入力することで直ることが多く，単純なわりに効果的な方法です。

以上はデバッグの基本中の基本ですが，多くのエラーはこれだけの作業で直ってしまいます。「答えはエラーメッセージにあり」と書きましたが，エラーメッセージのin×××の部分だけでも十分デバッグができるのです。

それでは，エラーの原因となるバグがエラー発生行とは異なる場合にはどうするかというと，ここでエラーメッセージの前半部分が生きてくるわけです。エラーメッセージの種類と意味はBASIC文法書（付属のマニュアル）に一覧表として載っていると思います。ただ，1つひとつの意味は書いてあっても実際にデバッグの仕方が書いてないと思いますので，ここでは具体的な方法をエラーメッセージ別に述べてみます。

その前にデバッグの基本としてたいへん重要なことを述べておかなければなりません。エラーが発生した行がマルチステートメントの場合，どの命令でエラーになったのかをはっきりさせる必要があるからです。

エラーメッセージからエラーが起こった命令を特定できる場合はそれでOKです。特定できない場合はどうすればよいか。いろいろ考えられると思いますが，意外に知られていない方法ですがマルチステートメントを2行に分けて書くのがたいへん有効な手段です。たとえば1500行でエラーになった場合，1500行の最初の命令だけ残して残りを1501行にします。これを走らせて1501行でエラーになったら1501行の最初の命令を1500行に移します。これを繰り返して1500行がエラーになったときに移された命令がエラー発生源であることがわかります。

2行以上に分けて書ける場合は，そうしたほうが早く見つかります。皆さんでいろいろ工夫してみてください。

それではいよいよエラーメッセージ別のデバッグマニュアルです。

NEXT without FOR	(13)
FOR without NEXT	
RETURN without GOSUB	(14)
Undefined label	(16)
UNTIL without REPEAT	
REPEAT without UNTIL	
WHILE without WEND	
WEND without WHILE	

※カッコ内はS-BASICのエラー番号

サンプル1　単純な打ち間違い

```
10 '<<九九の表>>
20 FOR I=1 TO 9
30   FOR J=1 TO 9
40     CURAOR 3*I,J
50     PRINT I*J
60   NEXT
70 NEXT
80 END
```

40行で「CURSOR」を「CURAOR」と打ち間違えたことにより「Syntax error in 40」

これらのエラーは本来対応して存在するはずのコマンドの一方が存在しないために起こるものです。ですから，FOR～NEXT, GOSUB～RETURN, REPEAT～UNTIL, WHILE～WENDの対応を考えて雑誌のリスト上でもう一方のコマンドがあるべき行番号を見つけ，その行のLIST（これ以後雑誌のリストを「リスト」，パソコン上でのリストを「LIST」と書くことにします）を見ると，その行自体を入力し忘れていたか，行は存在してももう一方のコマンドをその行の中に入力し忘れていたのを発見できるでしょう。しかし，リスト上で対応するもう一方のコマンドを発見するのはけっこう難しいもので，多少はBASICのことを知らないといけません。

また，LIST上の対応が正しくても，ほかからGOTO文などでループ内に飛び込んで来たためにこれらのエラーが起こることもあります。これは，そのGOTO文の飛び先の行番号（またはラベル名）が誤っていたためで，原因を見つけるのはなかなか難しいのですが，とりあえずリストとLISTでジャンプに関する命令(GOTO, GOSUB, IF)を中心にチェックしてみることです。または，プログラムの動きを追う（トレースする）方法もありますが，これは次のエラーメッセージグループでの説明の際に詳しく述べます。

そのほかにも，リストにはないのに余分に入力してしまったためにこのグループのエラーが発生することもあります。この場合余分に入力してしまった行で発生すれば問題ないのですが，正しいほうで発生することもあります。ですから，エラーの発生した行の前後のLISTを調べてみる必要もあります。

Undefined labelというエラーはGOTO命令などでジャンプする先の行番号やラベルがLIST中に存在しないときに発生します。これは，エラーの発生した行のジャンプ命令の飛び先の行番号の行がLIST中に存在するかどうかを調べてみます。

Illegal function call	(3)
Over flow	(2)
Subscript out of range	(3,7)
Division by zero	(2)
String too long	(5)

このグループをご覧になって，一体どうしてこれらのエラーメッセージが同一グループになるのか疑問に思われる方も多いと思います。これらがひとつにまとめられる理由というのは，デバッグの方法が同じようになるということです。ここで言うデバッグとは，最初に触れたとおりすべての原因は入力ミスであるという前提に立っているので，入力ミスの発見であると思ってください。すなわち「デバッグ＝リストとLISTで違っている個所を発見し訂正すること」です。当然のこととして，自分で作ったプログラムのデバッグにはそのまま適用できないかもしれませんが，今回の記事の目的である雑誌のリストをいかに正しく入力するかという点では非常に実戦的なノウハウだと思っています。

エラーの発生した行に入力ミスがある場合というのはすでに述べたとおりですので，ここでこのグループのエラーが出る原因というのは，変数に誤った値（文字列も含めて)が代入された，ということにつきます。ですから，デバッグとは，どこで誤った値が代入されているのか，また，誤った値が代入された原因は何で，それはどこの入力ミスによるのかを突き止めることです。しかし，BASICという言語の性質上，変数はグローバルなもの，つまり，プログラム全

サンプル2 ‖ FOR文を忘れた

```
10 '<<アスキーコードとキャラクタの対応表>>
20 FOR I=2 TO 15:CURSOR 3*I,0:PRINT I:NEXT I
30 FOR I=0 TO 15:CURSOR 3,I+1:PRINT I:NEXT I
40 FOR J=0 TO 15
50 C=I*16+J:CURSOR 3*I+1,J+1:PRINT CHR$(C)
60 NEXT J:NEXT I
70 END
```

40行で最初に「FOR I=2 TO 15」を入力し忘れたことによる「NEXT without FOR in 60」

サンプル4 ‖ 必要なENDを省略した

```
10 '<<分数の和（Cを負の数にすれば差になる）>>
20 PRINT "A/B+C/D=?"
30 INPUT "A=";A:INPUT "B=";B:INPUT "C=";C:INPUT "D=";D
40 E=A*D+B*C:F=B*D
50 GOSUB 80
60 PRINT A;"/";B;"+";C;"/";D;"=";E;"/";F
80 X=ABS(E):Y=ABS(F)
90 IF X<Y THEN Z=X:X=Y:Y=Z
100 Z=X-Y*INT(X/Y)
110 IF Z=0 THEN 140
120 IF Z=1 THEN RETURN
130 X=Y:Y=Z:GOTO 100
140 E=E/Y:F=F/Y:RETURN
```

70行の「END」を省略してしまったことにより「RETURN without GOSUB in 120」

サンプル3 ‖ NEXT文の入力ミス

```
10 '<<S=Σn>>
20 INPUT"N=";N
30 FOR I=1 TO N
40   S=S+I
50 NEXT J
60 PRINT "1+2+・・・+";N;"=";S
```

50行の「NEXT I」を「NEXT J」としたことによる「FOR without NEXT in 30」

サンプル5 ‖ 必要な注釈行を省略した

```
10 '<<日数計算（２つの日付の間の日数を計算する）>>
20 INPUT "START YEAR=";Y:INPUT "      MONTH=";M:INPUT "        DAY=";D
30 GOSUB 90:J1=J
40 INPUT "  END YEAR=";Y:INPUT "      MONTH=";M:INPUT "        DAY=";D
50 GOSUB 90:J2=J
60 J=J2-J1:IF ABS(J)<2 THEN PRINT J;"DAY":GOTO 80
70 PRINT J;"DAYS"
80 END
100 J=INT(365.25*Y)+INT(Y/400)-INT(Y/100)+INT(30.59*(M-2))+D:RETURN
```

90行の「'(REM)」を省略したことによる「Undefined label in 30」

サンプル6 GOTO文を忘れた

```
10 '<<二次方程式の解法>>
20 PRINT:INPUT "AX^2+BX+C=0       A=";A:IF A=0 THEN END
30 INPUT "                        B=";B
40 INPUT "                        C=";C
50 D=B^2-4*A*C
60 IF D<0 THEN 90
70 IF D=0 THEN 100
90 PRINT "X=";-B/2/A;"+-";SQR(-D)/2/A;"i":GOTO 20
100 PRINT "X=";-B/2/A:GOTO 20
110 PRINT "X=";-B/2/A;"+-";SQR(D):GOTO 20
```

80行で「GOTO 100」を入力し忘れたことによる「Illegal function call in 90」

サンプル8 配列宣言のミス

```
10 '<<三角関数表>>
20 DIM A(80):P=3.1415927
30 FOR X=0 TO 90
40   A(X)=SIN(P/180*X)
50 NEXT X
60 PRINT "x","sin(x)","cos(x)","tan(x)"
70 FOR X=0 TO 89
80   PRINT X,A(X),A(90-X),A(X)/A(90-X)
90 NEXT X
100 PRINT X,A(90),A(0),"-----"
110 END
```

20行で「A(90)」を「A(80)」としたことにより「Subscript out of range in 40」

サンプル7 演算子の入力ミス

```
10 '<<階乗計算>>
20 INPUT "N=";N:IF N>34 THEN 20
30 M=1
40 FOR I=1 TO N
50   M=M*I
60 NEXT I
70 PRINT N;"!=";M
```

20行で「N>=34」を「N>34」とじたため,「N=34」としたら「Overflow in 50」

サンプル9 変数名の打ち間違い

```
10 '<<平均・標準偏差(DATA=0で終了)>>
20 N=0:T=0:TT=0
30 INPUT "DATA=";X:IF X=0 THEN 60
40 T=T+X:TT=TT+X^2:M=N+1
50 GOTO 30
60 XA=T/N:SD=SQR(TT/N-XA^2)
70 PRINT:PRINT:PRINT " NUMBER=";N
80 PRINT "   TOTAL=";T
90 PRINT "AVERAGE=";XA
100 PRINT "      SD=";SD
110 END
```

40行で「N=N+1」を「M=N+1」としたことにより「Division by zero in 60」

体で共通であるために場所の発見が非常に難しくなることもありますので,ここで順序立てて述べたいと思います。

最初は,どの変数の値が誤っていたかを判別することから始めます。エラーの発生した行のLISTを表示させてその内容を検討します。そこで使われている変数が1つならば問題ありませんが,2つ以上の場合には,エラーメッセージの意味を考えてみます。

「Illegal function call」であればコマンドや関数の引数,パラメータが規定の範囲を越えているのですし,「Over flow」は数値の演算の際に起きます。「Division by zero」は0で割ると発生します(tan(π/2)でも起こるので注意)。また,「String too long」は文字変数の演算のとき(A$=B$+C$など)などに発生し,「Subscript out of range」は配列の添え字に宣言より大きい値を使用したりしたときに起こります。以上のような意味を考えると変数の特定がしやすくなります。

また,デバッグ(一般的にも)の常套手段として,変数の値をダイレクトモードで表示させてみるという方法もあります(PRINT A, B⏎などのようにする)。こうして,エラーの直接の引き金となった変数を見定めます。

次は,その変数にどこで誤った値が代入されたかを調べます。この方法としては,その変数の出てくる行をしらみ潰しに調べる(HuBASICではSEARCHを活用)ということがすぐに浮かびます。どこかの行の間違いが発見されるかもしれません。行の数が増えてくるとこの作業はかなり面倒になるでしょうが,ほかにこれといった方法もないようです。しかし,この方法で間違いが発見できないこともあります。たとえば,Aという変数が誤った値を持つとしてAの出てくる行はすべて正しくても,中にA=Bといった文があり,実はBのほうに誤った値が代入されていたという場合もあるからです。今度は,「A=」という代入文の右辺に出てくる文字にも注目していかなくてはいけません。こうなってくるとかなりの広範囲にわたってしまいますが,デバッグというのは多少は困難なものなので覚悟を決めてリストとLISTを見比べてください。

別の方法としては,トレースをしながら変数の値をチェックしていくというものがあります。先ほど触れたようにトレースについて説明します。ひとつにはBASICのTRONコマンド(もちろんTRONがあればですが)を使用する方法です。これを実行しRUNすると実行された行番号を表示しながらプログラムを実行するというものです。ただ,表示が速すぎて目で追えないときがあったり,画面表示が乱れてしまったりで,実際にはあまり役に立つとは言えません。

よく使われる手段はプログラム中の要所にSTOP文を書き,強制的に実行を停止して変数の値を表示させたりしながらチェックしていくというやり方です。こうしたテクニックは自作のプログラムのデバッグにも有効なのですが,STOP文の入れる位置やそのときに変数がどういう値をとっているべきなのかということをわかっていなければうまくいきません。したがって,多少はプログラミングの知識(リストの流れを理解していてBASICの文法がわかる)が要求されます。

以上のようにしていくといくつかのバグは発見できるでしょう。しかし,このように変数を追っていくというのは原理的には可能でも実際には膨大な手間がかかることがあり,もう一度最初から打ち直したほうが早いと思われるかもしれません。もし,万策尽きたらそれもよいでしょう。いずれにしても根気(根性)が必要です。

さて,変数(代入文により派生するものすべて,A=BではA,B両方とも)が出てくる行はまったく正しくてもまだエラーとなる場合もあります。これは,プログラムの流れを制御するコマンドのパラメータに誤りがある場合などです。特にループの回数の違いによるものが多く,たとえばFOR~NEXTで,10回繰り返すべきところを100回繰り返してしまうことなどによります。このために,このグループのエラーが発生

するというのはよくあることです。これは，プログラムのトレースをするとわかりますが，誤りのある変数を含むループの制御変数には特に注意してみるべきです。LISTをチェックする際に前後も確認します。

以上は，このグループ全般に関することです。ほかに「Subscript out of range」では，配列の大きさを定義(DIM文)している行に間違いがあるということも考えられます。

Out of data	(24)

このエラーメッセージが出るかどうかは別として，DATA文によるトラブルは非常に多いと言えるでしょう。ひとつには，DATA文というのは普通の場合，同じような形式のものが多くあり入力の際に間違いやすいということが理由でしょう。間違いの種類には，データの数が違っている（多いか少ない）ことと，データそのものが誤っているという2つがあります。

データの入力ミスがあった場合にはどういう現象が起こるかというと，PCG定義のデータやグラフィックのデータの誤りであれば表示されるキャラクタや画面が乱れるといったもので，動作自体には影響なくエラーメッセージの出ないバグとなります。そのほかはどういうことになるかは特定できず，ほとんどあらゆるエラーが起こり得るとも言えます。しかも，データの数が多い場合はたいへん誤りを発見するのが難しいという恐ろしいものです。ですからDATA文の入力は特に慎重に行う必要があります。

私は，データが非常に多いときには，リストを線で区切ったり定規をガイドに当てたりしながら入力し，見落としや自分が入力している場所を見失わないようにしています。前に書いた2人で入力するというのも，データの入力時には特に有効です。

さて，「Out of data」というメッセージですが，これはREAD文で読み込むデータが存在しないという意味で，READ文のある行で発生します。原因としては，上で述べたDATA文に誤りがある（数が足りない）ということと，READ文のほう(READ文のある行とは限らない）に問題がある場合の2つが考えられます。したがって，このエラーが発生した場合にはまずデータの個数をチェックします。個数が違えばどこかに入力し忘れたデータがあるはずです。個数がリストとLISTで一致していれば，一応原因はREAD文側にあると考えられます。中味が違う可能性はありますが，数さえ同じなら「Out of data」は起きません。

READ文側での誤りというのは，データの個数より読み込む回数が多かったのですから，LISTを見て何回読み込まれているかを調べます。この場合には,FOR～NEXTなどで読み込んでいることが多く，ループの制御変数(FOR I＝のIのこと)の値にも注意してみます。また，読み始めるDATA文の行番号を指定するのにRESTOREを使っていることもあり，この行番号が違っていればデータの個数が合っていてもエラーが出ることがあります。そのほかの原因ももちろん考えられますが，以上の点をチェックすることでもかなりバグの発見の可能性が高いと思います。

Out of tape	
Tape read error	
File not found	(40)
Device I/O error	(41)
Device full	(53)
Device off line	(50)
Write protected	(42)

これらは一見しておわかりになると思いますが，ハードのほうのトラブルによるものです。ですから操作ミス（違うテープやディスクを入れたり，プリンタのスイッチの入れ忘れや紙切れなど）によるものが大部分です。ただ，入力ミスによっても起こり得るのですが，その場合はエラーの発生した行には普通問題なく，その前の段階でファイルネームの設定やデバイスの設定でのミスが多いので注意が必要です。

そのほかのエラーメッセージ エラーメッセージ全般

以上取り上げたエラーメッセージのほかにもエラーメッセージはいくつかありますが，発生する頻度はそう高くないと思います。こう書くと有名な「Syntax error」がないではないかという反論があるかもしれません。しかし，これはエラー発生行に入力ミスがある場合に含まれますので，個別には取り上げませんでした。ただし，ユーザー定義関数を使っている場合,DEFFNの中に入力ミスがあっても呼び出したほうでエラーになりますので注意してください。

エラーメッセージが発生すると，「またか！」と嫌な気分になりますが，考えようによってはエラーの出るのはむしろラッキーな場合で，このあと述べるようにエラーすら出ないという恐ろしいバグもあるのです。上の説明を読んでもらうとわかるとおり，エラーが発生した場合にそれの原因を突きとめる特効薬というものは今のところありません。経験と直感に頼るしかしかたがないこともありますが，これまで述べた方法でもかなり役に立つと思います。「答えはエラーメッセージにあり」と言ったように，エラーメッセージの意味を十分に考えてデバッグしてください。

サンプル10 ‖ DATA文の入力ミス

```
10 '<<月別降水量>>
20 PRINT "   0         100        200        300(mm)"
30 PRINT "---+---------+---------+---------+-----"
40 FOR I=1 TO 12
50   READ A
60   X=INT(A/10+.5)
70   PRINT RIGHT$(" "+STR$(I),2)+" |";:FOR J=0 TO X:PRINT "*";:NEXT J:PRINT
80 NEXT I
90 PRINT "---+---------+---------+---------+-----"
100 DATA 54,63,102,128,148,181,125,137,193,181.93,56
```

100行で「181」と「93」の間の「，」を「．」としたことにより「Out of data in 50」

走ってはいるが動きがおかしい！

デバッグで困るのはむしろこちらのほうです。エラーは発生しないけれども説明のとおり動かないときには非常に困ってしまいます。しかし，せっかく入力したプログラムですから何とか完動させましょう。というわけで，この場合のデバッグの方法を考えてみます。

プログラムの中には，操作上のミスによるエラーを考慮してON ERROR文を使用しているものがあります。これは，エラーが発生した場合にエラー処理をするルーチンに飛ぶものですが，デバッグの際には邪魔なので，ON ERROR GOTO 0として無効にしておきます。ただ，エラーを故意に発生させてそれを利用するというハイテクニックを使ったプログラムもあるので万全の方法とはいえません。しかし，とりあえず実行してみてください。これでエラーが発生すれば，そのメッセージに即したデバッグを行っていけばよいのです。

また，先にやり方を説明したプログラムのトレースも有効な方法です。プログラムを少しずつ実行しながら正常に動く部分とそうでない部分を区別していきます。その際には変数の値のチェックも行います。

それからゲリラ的戦法としては，無意識の内に誰でも実践されていると思いますが，おかしな動きを始めたときにすぐBREAK(STOP)をかけて実行を止めてしまうことです。そして，止めた行の前後のLISTを詳しく見てリストとの違いを捜します。このためには，turbo BASICなどでBREAKを無効（BREAK OFF文）にするコマンドがあるので，これも最初にLIST中でREM文に変えておきます。

さて，動きがおかしい場合それに対処する基本的姿勢としては，プログラムの動き(現象)を分析していくことから始めます。難しい言い方になってしまいましたが，具体的にどんな動きをするかを見て，その原因を考えていくわけです。たとえば，簡単な例としては，音楽が流れてくるときの音階がおかしいのであればMUSIC（PLAY）文の中味が違っているだろうという予想ができます。キャラクタがおかしいのならPCG定義やDATA文に誤りがありそうですし，表示位置がおかしいのならばPRINT文やLOCATE(CURSOR)文のパラメータが違っているのです。

考えられるあらゆる手段を使ってもバグの所在を発見できないこともあります。この場合，行き着くところはLISTとリストをすべて比較していくことです。しかし，これも大変な作業で何とか避けたいものです。そこで，すべての行をチェックする前に一部分をチェックしてみるということを考えるわけです。具体的な方法としていくつか例を挙げてみます。

1)　数行おきに比較する

これは，入力後すぐに実行せず簡単な確認を行う際にも有効です。

2)　コマンド別にチェックする

SEARCH命令を使うと簡単ですが，ある特定のコマンドの含まれる行だけをチェックするという方法です。GOTO文やIF文FOR～NEXT文などがミスも起こりやすく，プログラムの流れにもろに影響を与える部分であるだけに効果的です。

3)　変数別にチェックする

2)と同じく特定の変数名の含まれている行だけをチェックするもので，変数の誤りによる異常はよくあることなのでこれも有効でしょう。

4)　DATA文をチェックする

エラーメッセージの「Out of data」の項で述べましたが，DATA文のミスは非常に犯しやすくて，しかもエラーが発生しないものが多いのです。キャラクタが乱れているとか，ペイントがはみ出すといったものののほとんどがデータの間違いからくるものです。したがって，DATA文のチェックによりバグを発見する可能性はかなりあります。

といったような方法がありますが，どうしてもバグが発見できない場合はすべての行をチェックすることになってしまいます。プリンタを持っている人はLISTを出力して，リストの載っているものとチェックする行を上下に並べて比較するというのも間違いを見つけやすくする方法です。

以上のように，デバッグというのは大変なものなのです。入力ミスの発見という意味だけでもこれだけのことがあり，まだまだ書き尽くせないことがたくさんあります。自作のプログラムのデバッグとなると，さらに別の要素（アルゴリズム上のバグ：考え方が誤っている）も加わるのでこの場では触れませんが，ここでの説明もいくらかは役に立つことがあると思います。また，投稿される方は，移植という点やこの入力後のデバッグという点を考えて，できるだけ第三者にもわかりやすいプログラムを書いてほしいと思います（特にメモリやスピードの上で余裕のある場合には）。

最後には根気がものをいうことになりますが，無事に入力できプログラムが動いたときの喜びはまた格別のものがあります。そのためにも，皆さん頑張ってデバッグしましょう。

質問電話は殺しのナンバー

「バグに関するお問い合わせ」の電話は別名「質問電話」と呼ばれている。ここで，我々が入手した「質問電話を有効に利用するための耳寄りな情報」を公開することにしよう。実行すれば電話代の節約にもつながるはずである。まずは予備知識から。

1)　答えられるのはあくまでも「バグに関するお問い合わせ」だけである。そのほかの質問，たとえば記事とは関係ないことやプログラムの移植・改造に関することには原則として答えてもらえない。

2)　1)と重なることであるが，「マニュアルをよく読んで勉強してください」としか答えようがない質問，たとえばマシン語入力の仕方などは，確かに記事とは関係あるが電話では答えようがないと思う。

ということでいよいよ問題の核心に入る。満足な回答を効率的に得るための必殺(?)テクニックだ。

1.　手元に本と筆記具を用意する

まず，年・月・ページ・記事を伝えよう。早く参照してもらうことができる。また，手元に本がないと話が通じにくく，せっかく聞いた話もあとでわからなくなってしまうことがあるので注意しよう。

2.　エラーの出る条件を伝える

何をどう操作したら異常が起きるのか，状況を詳しく伝えよう。エラーの原因を推定するための重要な手掛かりとなる。これを調べる段階で，使用するBASICが違うなどの基本的な入力・操作ミスを発見できることもあるだろう。

3.　エラーの内容を伝える

エラーメッセージの出るものは，行番号とエラー内容を伝えよう。できれば（本来ならば）メッセージの意味，エラーになったコマンド・変数を調べておきたい。これは次の3つの意味で重要なことである。

1)　自分自身でエラーの内容を確認することにより，電話をする以前に入力ミスやバグを発見できることがある。

2)　プログラムのバグではないことが明らかな場合でも，適切なアドバイスが受けられる可能性がある。

3)　雑誌が出て間もない時期は，担当者もなるべく詳しい情報を欲しがっているはずである。

以上述べたことの基本にあるのは，質問者，回答者ともに十分な情報を必要としている，ということである。回答者だってすべてお見通しというわけではなく，情報がまだ少ないときは，質問者がするのとまったく同じようにして考えている（ように見える)。違うのは情報が集まるところにいるかどうかだけだ。ここでも読者との共同作業が行われているのである。

もっともよくある入力ミスを担当者に聞いたら“データの打ち間違いによる「Illegal function call」,「Out of data」ですね”ということであった。質問電話を担当者の殺しのナンバーではなく，バグ殺しのナンバーとして活用したいね！　（中川 智哉）

他機種の記事を活用する

BASICプログラム移植の心得

Kudo Makoto
工藤 誠

毎月たくさんのプログラムが雑誌に発表されています。その中から自機種用のものしか利用しないなんてもったいないですね。ほんのわずかの変更で走るものがいっぱいあるのですから。というわけで，Oh！MZ移植担当者が語る「プログラム移植の心得」です。

移植というのは，あるパソコンで走るソフトをほかのパソコンで走るように手直しすることである。

市販のゲームなどは，マシン語レベルでの移植を行っているわけだが，これは，各機種のハードに深くかかわってくるので，たいへん複雑な作業となる。しかし，BASICの移植となると，一応BASICという特定の言語の上での話なので，多少は簡単になるのである。それでは，まず最初に，移植を始める前の準備としてどんなことをしたらよいかをこれから述べることにする。

移植作業の準備

1．なにはともあれBASICの基礎は覚えよう。

これは常識である。だが，あまりBASICを知らない人でも，移植が可能なことだってある。あとで述べるが，移植が比較的容易なプログラムと，そうでないプログラムがあるのである。

2．自分の使っているパソコンで何ができるかをしっかり把握しよう。

BASICの性能や，マシンのハードによって，どうしても移植が不可能なプログラムもある。たとえば，MZ-700でグラフィックを扱おうとするのは，素人が素手でロードウォリアーズと戦おうとするようなものである。最初からあきらめて相手にしないほうがよい。もっとも，プログラムのアルゴリズムだけを使って，最初から新しくプログラムを作り直せばよいかもしれないが，とても簡単な移植とは言い難い。

3．暇があったら，他機種のBASICの特有の命令も知っておこう。

これは別に必ずしも必要なことではない。BASICの比較表がひとつあれば十分である。ただ，知っておくと移植作業がはかどるのである。

移植というのは，とにかく元になるプログラムによって，難しさや可，不可が決まってしまうのである。したがって以上の3つの条件などから「とても僕には無理だ」という人でもプログラムによっては移植ができてしまうのである。ではまず例として，次のHuBASICで書かれたプログラムをS-BASICに移植してみよう。

```
10 LOCATE 0,0
20 PRINT "ABC"
```

これは，LOCATEをCURSORに直せばよいのである。簡単なことでしょ。さあ，自信がついたところで先に進もう。

技術面では以上の条件を満たしていればよい。問題なのは，精神面である。移植作業はテキストコンバータなどを使わない限り，原則としてプログラムを自分で打たなければいけない。しかも，ただリストを打つだけではなく，ステートメントや関数などが，自分の機種で使えるか，他の命令で置き換えるかなどと頭を働かせたり，資料を調べたりしなくてはいけない。不屈の精神，雑草の根性，燃える闘魂が必要なのである。移植のできない人とできる人には決定的な違いがある。それは「執念」である。まあ，簡単に言うと，やる気があれば，誰にでもできるということなのだ。やる気さえあれば，多少の知識の不足も問題ではなくなる。では，準備ができたところで次に進んでみよう。

移植の実際

シャープのパソコンは，大まかにMZ系とCZ系に分類され，それぞれにタイプの違うS-BASICとHuBASICが使われている。おのおののBASICの間では，命令語が似かよっているので，ほとんど移植の手間はかからない。ということで，S-BASICと，HuBASICの間の移植を中心に述べることにする。

まずS-BASICからHuBASICへの移植を考えてみよう。ハードに深くかかわってくるステートメントなどを除くと，S-BASICのステートメントや関数は，ほぼHuBASICでカバーできる。すなわち，同じものがあるか，あるいは代わりになるものがあるのだ。逆にHuBASICで作られたプログラムをS-BASICに移植しようとすれば，大変な手間がかかってしまう。S-BASICにない関数などは，ほかのステートメントを組み合わせてプログラムを組まなくてはならない。私はここでBASICの良し悪しを論ずるつもりはないが，移植に関する限り，ステートメントや関数が多いほうが有利であると考えている。そんなわけで，HuBASICからS-BASICへの移植の話が自然と多くなってしまった。

移植のタイプは，大きく4つに分けられる。やさしい順に説明してみよう。

タイプ1――名前，書式，働きがまったくあるいはほぼ同じもの。

たとえば，PRINTなどがこれに当たる。FOR～NEXTや，IFの条件文などもこの

タイプであろう。言うまでもないが，このタイプのステートメントや関数は移植が楽である。これも移植のうちに入るのであろうか，という疑問を抱きつつ，次のタイプに行ってみよう。

タイプ2——名前，書式が違っていても，ほかのステートメントや関数で代用可能であるもの。

ようやく移植らしくなってきた。例を挙げると，先ほど述べたCURSOR↔LOCATE，また，PRINT/P↔LPRINT などがこのタイプに入る。このタイプの移植もまた楽である。要するに，各 BASIC の比較表なんかがあれば，簡単に書き換えて移植することができる。今月から3回にわたって掲載される「BASIC DATA LIST」をおおいに活用しよう。

以上2つのタイプのステートメントや関数のみに使っているプログラムは，あまりBASICのことがわからないような人でも移植することが可能である。アルゴリズムを理解する必要もないし，プログラムを解析する必要もない（ただし，変数名の制限などの問題は残っているが）。しかし，実際にこんな単純なステートメントだけで書かれているプログラムには，めったにお目にかかることはない。普通は，次のタイプ3が移植作業のメインとなっている。

タイプ3——代用できるステートメント，関数がなく，プログラミングして同じ働きをさせなければならないもの。

具体的な例は，あとでリスト付きで説明するが，ここまでくると，もはやBASICの知識なしには移植できなくなってくる。特に，命令数の少ないBASICでは，このタイプの移植が多くなりがちである。この話はあとに譲って，次のタイプに進んでみよう。

タイプ4——BASIC 上では，移植が不可能なもの。

BASIC上で移植が不可能と言っても，マシン語レベルでは可能なものがある。たとえばMZ-80B/2000/2200では，

POKE＄952，0：GETK＄：＄952，166

とすれば，リアルタイムキー入力（HuBASICでは，K＄＝INKEY＄（0））が可能になるし，

USR（＄0F14）

で，BEEP音を発することができる。

一般的にはこのようなものではなく，根本的にハードがサポートできないなどの理由によるものが多い。しかし私は，このタイプに対する必殺テクニックを2つばかり持っている。それを今回は，特別に伝授することにする。

第1のテクニックは，すっぱり移植を諦めることである。これは，あまり説明を要しないだろうし，私もまた多くは語りたくない。第2のテクニックは，そのステートメントなどを使っている部分を，ばっさり切り捨てることである。たとえばゲームなどで，ゲームオーバーの際に3重和音の音楽を流す部分など，プログラム全体の中で本筋にあまり影響を与えないところは，機能を大幅にカットするか，思い切って切り捨てるべきである。何も完璧な移植を行う必要はないのだ。このテクニックはかなり有効であるが，多用しすぎるとプログラムが消えてしまうという欠点がある。程度をわきまえて使ってほしい。

さて，ほとんどのコマンド，ステートメント，関数，システム変数などは，以上の4つのタイプに分類できるのだが，これらの中に含まれないものもないわけではない。すなわち，名前や書式が同じ，あるいは非常に似ていても，処理が違うという悪夢のようなタイプである。代表的なものを，4つ挙げておこう。

ひとつはLOGという関数である。これはS-BASICでは，常用対数すなわち $\log_{10}$ を示しているのだが，HuBASICでは，自然対数 ln($\log_e$) を示しているのである。これなどは，思わずタイプ1と間違えそうになるであろう。

HuのLOG（X）↔SのLN（X）

SのLOG（X）→HuのLOG（X）/LOG（10）

これでよい。対数ってなあに？　とおっしゃられる読者の方も，何も考えずこのように変換すればよろし。

次の関数はRND（X）である。HuBASICの場合Xはダミーなのでなくてもよく，乱数系列の設定はRANDOMIZE 命令で行う。それゆえ，Xはどんな数でもよい。ところがS-BASICでは，Xが0以下の数で乱数系列の初期化が行われるので，HuからSへの移植の際にXが0になっているとき，ああ，こりゃ同じ命令だなんて書き換えずに打ち込むと，その時点でもうすでに死んでいるのである。

X＞0:SのRND（X）↔HuのRND（X）

X≦0:SのRND（X）↔Huでは，RANDOMIZE(n)命令を使う。

逆に，HuBASICのRANDOMIZEをS-BASICにするのは，システム変数 TI＄ を使って以下のようにするとよいだろう。

```
10 T=VAL(RIGHT$(TI$,2))
20 FOR I=0 TO T
30 R=RND(1):NEXT
```

乱数はゲームなどでは大変重要な要素であるから，おろそかにしてはいけない。と，これで乱数もよろし。

最後はシーケンシャルファイルへの出力である。これは，名前が多少違っているが，ファイルをオープンして，PRINT 命令で書き込んで，ファイルをクローズする処理は同じように見える。移植のタイプから言えば，先に述べた中のタイプ3，代用命令がある類ではないかと思うだろうが，実はこれは罠なのである。

ファイルへの出力は，PRINT＃nあるいはPRINT/Tなどで行われるが，複数のデータをカンマ“,”で区切って出力した場合の処理が，HuBASICとS-BASICでは異なっている。HuBASICは，スクリーンに出力するのと同じようにファイルに出力するので，カンマで区切っても複数のデータとしては扱わないのだが，S-BASICの場合はきちんと複数のデータとして扱う。つまり，S-BASICの「PRINT＃n(PRINT/T) A＄，B＄，……」は，HuBASICにすると，「PRINT＃n A＄:PRINT＃n B＄:PRINT＃n……」となることに注意しなければならない。なお，言うまでもないことだが，A＄，B＄が数値データ（たとえばA，B，10など）でもかまわない。最後になったが，気を付けなくてはならないのは，キャラクタの相違である。グラフィックキャラクタは言うまでもない。同じキャラクタでも，アスキーコードが違うものがあるので注意すべし。CHR＄（～）とASC（～）の関数があったら，そこで使われるデータは十分調べる必要がある。コントロールコードなどの働きも，各機種間で相当に違っている。たとえば，X1で画面をクリアするとMZ-2000で同様のことをするのでは次のような違いがある。

X1　PRINT CHR＄（12）

2000　PRINT CHR＄（6）

各機種のアスキーコードとコントロールコードを表1，表2に示しておく。

これで，実際にどうやって移植するのか，雰囲気だけでもわかってもらえたであろう。まさに移植作業は，人間業とは思えないくらいの手間と時間がかかるのである。

サブルーチン集

ここで取り上げるサブルーチンは，HuBASICからS-BASICの移植を行う際に，代用する命令がない関数や，ステートメントをプログラミングしたものである。当初は，S-BASICからHuBASICへの移植に役立つサブルーチンも挙げるつもりだったのだが，S-BASICの主な命令は，すべてHuBASICで同じものや，代わりになるものが見つかったりする。つまり，先ほどの移植のタイプで言えば，タイプ1と2で，BASICの比較表があれば，誰でも移植できるのである。そこで，HuBASICからS-BASICへの移植サブルーチン集になったわけだ。プログラムは，初心者が理解しやすいように，ということを考えて作ったので速度などの点は良いプログラムとは言えない。実際に使うときは，自分なりに改造して使っていただきたい。なお，正しくない引数に対する処理は，ほとんど何もしていないので，これらのサブルーチンを呼ぶときには変数の値に十分な注意が必要となる。では，個個のリストの説明に移ろう。

リスト1　INKEY＄(1)

この関数は，カーソルを出してキー入力があるまで待ち，キー入力があったらその入力されたキャラクタを返すものである。A＄にカーソル用のキャラクタと，カーソルを左に移動させるコードを入れておく(MZ-80B/2000はCHR＄(4)，MZ-80K/C/1200ではCHR＄(20))。B＄には，A＄のカーソル用キャラクタの代わりにスペースを入れ，A＄とB＄を交互に画面に出力することによってカーソルのように見せるのである。入力されたデータはIN＄という変数に入り，サブルーチンから帰ってくる。

リスト2　INSTR(B＄，A＄)

B＄の文字列の中で，何番目にA＄が入っているかを捜す関数で，もしA＄が見つからない場合は0を返してくる。A＄，B＄それぞれ値を決めてこのサブルーチンを呼ぶと，数値変数INにINSTRの値が入るようになっている。なお，B＄のn番目から捜す場合(HuBASICではINSTR(n，B＄，A＄))は，1020行のFOR～NEXTの初期値を1ではなくnにすればよい。この関数は，アドベンチャーゲームや人工知能などの会話型プログラムを作る際に使うと便利である。

リスト3　HEXCHR＄(X＄)

X1やMZ-1500では，データを16進数で扱うことが多い。これはキャラクタパターンの定義に使われるのだが，グラフィックパターンの定義にも16進数は便利である。このHEXCHR＄という関数は，16進数の文字列たとえば“313245”などを2つずつアスキーコードと考えて対応する文字列“12A”に変えるものである。X＄に16進数の文字列を入れてサブルーチンを呼ぶと，HC＄に答えが入って返ってくる。

リスト4　HEX＄(X)

数値を16進で表したいとき，たとえばX1ではディスクの内容をダンプするプログラムなどに使ってあるが，この関数は便利である。数値変数X（ただし－65535＜X＜65535)に値を入れると，HE＄に16進数の文字列が入る。

リスト5　VAL(“&B”＋X＄)

X1 HuBASICでは，16進数，8進数のほかに2進数も扱えるようになっている。8進数はともかく，2進数はグラフィックのパターンが一目見ればわかるという利点を持つ。2進数“0110…”などの文字列は，HuBASICでは“&B”をつけてVAL関数を使えばすぐ10進数になる。そこでこれをS-BASICで行ったのがこのプログラムである。X＄に2進数を入れサブルーチンを呼ぶと，Bに値が入る。なお，1080行にあるように，Bの値には制限があり，X＄の長さも16桁までである。

リスト6　キャラクタのBF

キャラクタを使って四角形を描き，内部も埋めるプログラムである。キャラクタでラインを引くのは，あまり使われないが，四角形はよく使われる。X0，Y0，X1，Y1，で四角形の対角位置を決め，A＄にセットした1文字のキャラクタで四角形を描くようになっている。X1F など NEW BASICで走らせるときは，1030～1060行，1090行

リスト1　INKEY$(1)

```
1000 'LIST 1
1010 'IN$=INKEY$(1)
1020 IN$=""
1030 A$="▒■":B$=" ■"
1040 PRINT A$;
1050 FOR I=1 TO 50:GET IN$:IF IN$<>"" THEN RETURN
1060 NEXT:PRINT B$;
1070 FOR I=1 TO 50:GET IN$:IF IN$<>"" THEN RETURN
1080 NEXT:GOTO 1040
```

リスト2　INSTR(B$，A$)

```
1000 'LIST 2
1010 'IN=INSTR(B$,A$)
1020 FOR I=1 TO LEN(B$)-LEN(A$)+1
1030  IF MID$(B$,I,LEN(A$))=A$ THEN IN=I:RETURN
1040 NEXT I
1050 IN=0:RETURN
```

リスト3　HEXCHR$(X$)

```
1000 'LIST 3
1010 'HC$=HEXCHR$(X$)
1020 HC$=""
1030 FOR I=1 TO LEN(X$)-1 STEP 2
1040  H=0
1050  FOR J=0 TO 1
1060   A=ASC(MID$(X$,I+J,1))
1070   IF A>64 THEN A=A-55:GOTO 1090      '... A - F
1080   A=A-48                             '... 0 - 9
1090   H=H+A*(16-J*15)
1100  NEXT J
1110  HC$=HC$+CHR$(H)
1120 NEXT I
1130 RETURN
```

リスト5　VAL(“&B”＋X$)

```
1000 'LIST 5
1010 'B=VAL("&B"+X$)
1020 B=0
1030 IF LEN(X$)>16 THEN RETURN
1040 FOR I=LEN(X$) TO 1 STEP -1
1050  A=ASC(MID$(X$,I,1))-48              '... A = 0 or 1
1060  B=B+A*(2^(LEN(X$)-I))
1070 NEXT I
1080 IF B>32767 THEN B=B-65536!           '-32769 < B < 32768
1090 RETURN
```

リスト4 HEX$(X)

```
1000 'LIST 4
1010 'HE$=HEX$(X)
1020 HE$="":H=X
1030 IF H<0 THEN H=H+65536!
1040 FOR I=3 TO 0 STEP -1
1050  A=16^I
1060  H0=INT(H/A):H=H-H0*A
1070  IF (H0<=0)*(HE$="") THEN GOTO 1100
1080  IF H0<10 THEN HE$=HE$+CHR$(48+H0):GOTO 1100        '... 0 - 9
1090  HE$=HE$+CHR$(55+H0)                                 '... A - F
1100 NEXT I
1110 RETURN
```

を削って，1064，1095行を復活させればよい。2000，2200の場合は，1030～1060行を削って，1066行を復活させる。ただし，このサブルーチンを使って，スクリーンの一番右下（CURSOR39, 24あるいは79, 24）に四角形の一端がかかる場合，画面がスクロールすることもあるので注意が必要である。なお，四角形の中を埋めずに，枠だけを残す場合は，四角形の中をスペースなどで消せばよい。

リスト7　CIRCLE(X, Y), R

これはおまけである。グラフィックは使えるのにサークルは書けないというS-BASIC MZ-1Z001を使っている人のために，円を描くプログラムを作ってみた。中心座標をX，Yに，半径をRに入れて呼べばよい。円を描くのがやたらに遅いX1 HuBASICよりも，もしかしたら速いかな？　と思ってHuBASICでもやってみたが，やはりCIRCLE命令の方が速かった。1070行を，

LINE(X0, Y0)−(X1, Y1), PSET, 7

に変えるとHuBASICでも走る。もの好きな人はやってみてください。

リスト6 キャラクタのBF

```
1000 'LIST 6
1010 'LINE(X0,Y0)-(X1,Y1),A$,BF
1020 IF LEN(A$)>1 THEN RETURN
1030 B$=""
1040 FOR I=1 TO ABS(X0-X1)+1
1050  B$=B$+A$
1060 NEXT I
1064 'B$=STRING$(ABS(X0-X1)+1,A$)
1066 'B$=STRING$(A$,ABS(X0-X1)+1)
1070 X=X0:IF X1<X0 THEN X=X1
1080 FOR Y=Y0 TO Y1 STEP SGN(Y1-Y0)
1090  CURSOR X,Y
1095 'LOCATE X,Y
1100  PRINT B$;
1110 NEXT Y
1120 RETURN
```

リスト7 CIRCLE(X, Y), R

```
1000 'LIST 7
1010 'CIRCLE(X,Y),R
1030 X0=X+R*COS(0):Y0=Y+R*SIN(0)
1040 FOR I=0 TO 6.4 STEP .3
1050  X1=X+R*COS(I):Y1=Y+R*SIN(I)
1060  LINE X0,Y0,X1,Y1
1070  X0=X1:Y0=Y1
1080 NEXT I
```

移植しやすいプログラムとは

これらのリストを見て，BASICのわかっている人なら，「何でー簡単じゃねーか」などと思っているだろう。そうなのだ，移植という作業は，技術的にはそれほど高度な知識を必要としない。移植担当の私本人が言うのだから間違いはない。問題なのは根気なのである。とにかく移植作業は苦労が絶えない。

そこで人一倍なまけものの私としては，仕事が楽になる方法を思いついたのである。つまり，読者の皆さんが，私が移植しやすいようなプログラムを作ってくだされればよいのだ。というわけで，これから私は，いかにしたら移植しやすいプログラムが作れるかを述べることにする。ここだけはしっかり読んでいただきたい。

その1　わかりやすいプログラムを！

わかりやすいプログラムとは何か？　詳しく言うと，プログラムが見やすく，アルゴリズムが簡単にわかるようなプログラムである。具体的にどうするかと言うと，FOR～NEXTのループの中は見やすいように1文字下げる，わずか1行中に命令をたくさんつめ込むなどということはやめる，注釈を効果的に使う，などなど。とにかく人に見せても恥ずかしくないようなプログラムを作っていただきたい。

その2　サブルーチンを使って！

たとえば書式の違うグラフィック命令などが，プログラムのあちらこちらに多数ちらばっていたらどうなるであろうか。その命令ごとに書き直すのはたいへんに疲れるのである。これが，たとえばラインはラインのサブルーチン，サークルはサークルのサブルーチンとまとまっていれば，そのサブルーチンだけを直せばすむのである。各機種間で異なる命令を何回も使うときは，BASICでサブルーチンが使えるということを思い出してほしい。

その3　特有の命令はなるべく使わない！

その命令を使わなければ，絶対にこのプログラムは成立しない！　と力を込めて言える場合ならばかまわないが，それ以外のときはなるべく機種特有の命令を使わないでいただきたい。どこかのソフトをまねて，意味もないのにTVを消したりつけたりするなんてことはじないように。プログラム全体から見て意味がないなら，特有な命令は自分で切ってもらいたいと思う。書き換えるだけでも手間がかかる移植作業に，よけいな作業が増えてはたまったもんじゃない。また，各機種に特有のキャラクタもできるだけ使わないほうがよい。

実際にあったプログラムの話だが，文字変数に意味のないキャラクタを入れる部分があった。プログラムを解析してみると，その文字列の1文字ごとのアスキーコードをデータとして使っているのである。まるで私にケンカを売っているようなプログラムであった。数値データだったら数値を使えばいいのに……。また，ステートメントで代用できるのにコントロールコードを使って，

PRINT CHR$(12)

などと書くのも困りものである。いつまでもPC-80の影を引きずっていてはいけない。コントロールキャラクタならわかりやすいので使ってもよいが，MZ系だけである。X1では，コントロールキャラクタをプリンタで印字しようとすると誤動作の原因となるので絶対に使わないように。

その4　どうして，各BASICすべて共通の命令だけを使わないのダ！

失礼しました。ちょっと手がすべって本音を書いてしまった。しかしよく考えてみると，そんなプログラムばかりになると，移植作業は必要なくなり，移植担当の私は失業してしまうではないか。これは困った。というわけで，その4だけは忘れていただきたい。失業はしたくないが，作業は楽をしたい。人間はなんとぜいたくなのであろうか。ただ，これだけは言っておきたいが同じような内容のプログラムなら，移植しやすい方が雑誌に載る可能性が高い。思いやりの気持ちを，プログラミングまで生かしてほしいなあ，と思うのであった。

表1　各機種のコントロールコード一覧

コード		X1, MZ-700(HuBASIC), MZ-2500(PC, FMもほぼ同じ)		MZ-700(S-BASIC) MZ-1500		MZ-80B/2000/2200	
10進	16進	処理内容	キー	処理内容	キー	処理内容	キー
1	01	インサートモード				カーソル下移動	↓
2	02	1ワード分左に戻る				カーソル上移動	↑
3	03	BREAKする (X1, MZ-700)	SHIFT＋BREAK	ひらがなモードにする (700)	CTRL＋カナ	カーソル右移動	→
4	04	スクリーンなどの初期化				カーソル左移動	←
5	05	カーソルから右を行の終わりまで消去		小文字あるいはひらがなにする		カーソルをホーム位置に移動	HOME
6	06	1ワード分右に進む		大文字あるいはカタカナにする		画面消去	CLR
7	07	ベルをならす				1文字消去	DEL
8	08	1文字消去	DEL			空白挿入	INST
9	09	水平TABの実行	TAB			グラフィック↔ノーマルモード切り替え	
10	0A	行の分離, ラインフィード				シフトロック↔ノーマルモード切り替え	
11	0B	カーソルをホーム位置へ移動	HOME				
12	0C	画面消去	CLR			カナ↔ノーマルモード切り替え	
13	0D	キャリッジリターン	CR, ↵	キャリッジリターン	CR	キャリッジリターン	CR
14	0E	カーソルより上を上方向にスクロール				ノーマルモードにする	
15	0F	カーソルより下を下方向にスクロール				カナモードをキャンセルする	
16	10	大文字あるいはひらがな(700)		1文字消去	DEL		
17	11	※1		カーソル下移動	↓		
18	12	空白挿入	INS, INST	カーソル上移動	↑		
19	13	※2		カーソル右移動	→		
20	14	水平 TAB の設定		カーソル左移動	←		
21	15	英数モードにする(MZ-700)	英数	カーソルをホーム位置へ移動	HOME		
22	16	カナモードにする(MZ-700)	カナ	画面消去	CLR		
23	17	次の行と結合(X1, MZ-2500)		グラフィックモードにする	GRAPH		
24	18	※3		空白挿入	INST		
25	19			英数モードにする	英数		
26	1A	カーソル以下をすべて消去		カナモードにする	カナ		
27	1B		ESC	BREAKする	SHIFT＋BREAK		
28	1C	カーソル右移動	→	ひらがなモードにする (1500)	CTRL＋カナ		
29	1D	カーソル左移動	←				
30	1E	カーソル上移動	↑				
31	1F	カーソル下移動	↓				

※1 一時停止の解除(X1)
　小文字あるいはカタカナ(MZ-700)
※2 一時停止(X1)ひらかなモード, CTRL＋カナ(MZ-700)
※3 カーソル行をすべて消去(MZ-2500)次の行と結合(MZ-700)

表2　各機種アスキーコード一覧

X1/X1turbo

16進	2進	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000				0	@	P	`	p	＿	─		ｰ	タ	ミ	●	▒
1	0001			!	1	A	Q	a	q	▬	│	。	ア	チ	ム	○	土
2	0010			″	2	B	R	b	r	▬	┴	「	イ	ツ	メ	♠	金
3	0011			#	3	C	S	c	s	▬	┬	」	ウ	テ	モ	♥	木
4	0100			$	4	D	T	d	t	■	┤	、	エ	ト	ヤ	♦	水
5	0101			%	5	E	U	e	u	■	├	・	オ	ナ	ユ	♣	火
6	0110			&	6	F	V	f	v	■	┼	ヲ	カ	ニ	ヨ	◢	月
7	0111			'	7	G	W	g	w	█	┐	ァ	キ	ヌ	ラ	◣	日
8	1000			(	8	H	X	h	x	▏	┘	ィ	ク	ネ	リ	╳	時
9	1001			)	9	I	Y	i	y	▎	└	ゥ	ケ	ノ	ル	▘	分
A	1010			*	:	J	Z	j	z	▍	┌	ェ	コ	ハ	レ	▖	秒
B	1011			+	;	K	[	k	{	▌	╮	ォ	サ	ヒ	ロ	▝	年
C	1100			,	<	L	¥	l	│	▋	╰	ャ	シ	フ	ワ	▗	円
D	1101			-	=	M	]	m	}	▊	╯	ュ	ス	ヘ	ン	▞	人
E	1110			.	>	N	^	n	￣	▉	╭	ョ	セ	ホ	゛	▚	生
F	1111			/	?	O	_	o	π	╱	╲	ッ	ソ	マ	゜	□	〒

MZ-80B
2000
2200

16進	2進	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000				0	@	P	`	p	‖	＝	■	■	タ	ミ	■	■
1	0001	⇩		!	1	A	Q	a	q	↓	¥	。	ア	チ	ム	■	■
2	0010	⇧		″	2	B	R	b	r	↑	■	「	イ	ツ	メ	■	■
3	0011	⇨		#	3	C	S	c	s	→	●	」	ウ	テ	モ	■	■
4	0100	⇦		$	4	D	T	d	t	←	○	■	エ	ト	ヤ	■	■
5	0101	Ⓗ		%	5	E	U	e	u	♠	┐	■	オ	ナ	ユ	■	■
6	0110	Ⓒ		&	6	F	V	f	v	♥	┘	ヲ	カ	ニ	ヨ	■	■
7	0111			'	7	G	W	g	w	♦	┌	ァ	キ	ヌ	ラ	■	■
8	1000			(	8	H	X	h	x	♣	└	ィ	ク	ネ	リ	■	■
9	1001			)	9	I	Y	i	y	╞	┼	ゥ	ケ	ノ	ル	■	■
A	1010			*	:	J	Z	j	z	╡	│	ェ	コ	ハ	レ	■	■
B	1011			+	;	K	[	k	{	╥	─	ォ	サ	ヒ	ロ	■	■
C	1100			,	<	L	\	l	│	╨	┴	ャ	シ	フ	ワ	■	■
D	1101			-	=	M	]	m	}	╫	┬	ュ	ス	ヘ	ン	■	■
E	1110		█	.	>	N	^	n	~	╪	┤	ョ	セ	ホ	゛	■	■
F	1111		▒	/	?	O	_	o	↰	╬	├	ッ	ソ	マ	゜	■	π

MZ-80K/C
1200
700
1500

16進	2進	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000				0	@	P	[illegible]	日	↓	─	タ	ミ	→	┌	─	┐
1	0001		⇩	!	1	A	Q	[illegible]	月	。	ア	チ	ム	▌	┴	♠	●
2	0010		⇧	″	2	B	R	[illegible]	火	「	イ	ツ	メ	▬	┬	│	＿
3	0011		⇨	#	3	C	S	[illegible]	木	」	ウ	テ	モ	￣	┤	─	♥
4	0100		⇦	$	4	D	T	[illegible]	水	、	エ	ト	ヤ	＿	▎	─	│
5	0101		Ⓗ	%	5	E	U	[illegible]	金	・	オ	ナ	ユ	│	▌	￣	◢
6	0110		Ⓒ	&	6	F	V	¥	土	ヲ	カ	ニ	ヨ	▒	▌	＿	╳
7	0111			'	7	G	W	☻	生	ァ	キ	ヌ	ラ	│	▀	│	○
8	1000			(	8	H	X	☺	年	ィ	ク	ネ	リ	█	▀	│	♣
9	1001			)	9	I	Y	[illegible]	時	ゥ	ケ	ノ	ル	◤	▄	◣	│
A	1010			*	:	J	Z	[illegible]	分	ェ	コ	ハ	レ	▌	┘	╰	♦
B	1011			+	;	K	[	[illegible]	秒	ォ	サ	ヒ	ロ	├	╭	╯	┼
C	1100			,	<	L	\	[illegible]	円	ャ	シ	フ	ワ	◘	╮	└	[illegible]
D	1101			-	=	M	]	[illegible]	¥	ュ	ス	ヘ	ン	└	┘	╲	│
E	1110			.	>	N	↑	[illegible]	£	ョ	セ	ホ	゛	┐	▞	╱	◥
F	1111			/	?	O	←	÷	[illegible]	ッ	ソ	マ	゜	▄	▚	┌	π

MZ-2500
(PC, FMもほぼ同じ)

16進	2進	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000				0	@	P	`	p	＿	┴		ｰ	タ	ミ	＝	╳
1	0001			!	1	A	Q	a	q	▬	┬	。	ア	チ	ム	╞	円
2	0010			″	2	B	R	b	r	▬	┤	「	イ	ツ	メ	╪	年
3	0011			#	3	C	S	c	s	▬	├	」	ウ	テ	モ	╡	月
4	0100			$	4	D	T	d	t	■	￣	、	エ	ト	ヤ	◢	日
5	0101			%	5	E	U	e	u	■	─	・	オ	ナ	ユ	◣	時
6	0110			&	6	F	V	f	v	■	│	ヲ	カ	ニ	ヨ	◥	分
7	0111			'	7	G	W	g	w	█	▕	ァ	キ	ヌ	ラ	◤	秒
8	1000			(	8	H	X	h	x	▏	┌	ィ	ク	ネ	リ	♠	〒
9	1001			)	9	I	Y	i	y	▎	┐	ゥ	ケ	ノ	ル	♥	市
A	1010			*	:	J	Z	j	z	▍	└	ェ	コ	ハ	レ	♦	区
B	1011			+	;	K	[	k	{	▌	┘	ォ	サ	ヒ	ロ	♣	町
C	1100			,	<	L	¥	l	│	▋	╭	ャ	シ	フ	ワ	●	村
D	1101			-	=	M	]	m	}	▊	╮	ュ	ス	ヘ	ン	○	人
E	1110			.	>	N	^	n	￣	▉	╰	ョ	セ	ホ	゛	╱	[illegible]
F	1111			/	?	O	_	o	[illegible]	┼	╯	ッ	ソ	マ	゜	╲	

BASIC DATA LIST

他誌を10倍楽しむ方法

習うより慣れろ。――何度となく口に，耳にするこの言葉ですが，それだけ的を得たものなんでしょうね。ところでそのBASICですが，いくら慣れろと言ってもそう次から次へと新しいアイデアが出てくるわけには参りません。何か手近な題材はなどと考えるほど，何も浮かばなかったりするのです。かと言って人のプログラムを打って動いた，というだけでは慣れたことにはならないし。

というわけで，ここはひとつ，他誌に掲載されたプログラムの移植にチャレンジしてみてはいかがでしょうか。うまくいった暁には，そのプログラムを使って楽しめてしまう，という，一大特典も付いています(当然)。おまけに相手のBASICもわかってしまうのですから，一石三鳥とはまさにこのことなのです。これら基本BASICのDATA LISTを，今月から3回にわたり連載します。このリストをいかに活用するかはあなたの努力しだいです。それではドーンといってみましょうか。（風間 浩)

表1 BASIC DATA LIST活用にあたっての注意点

行番号	ラベルの使えるBASIC，つまりN88，Hu，M25，S25，S(②③)では，原則としてラベルを使うこともできる。ただし，Hu，S(②③)では主にプログラム中でのみ使用可能（LISTやRUN行番号ではファイル指定との区別がつかないので使用不可)。
範囲	開始行番号［－〔終了行番号〕］または－終了行番号を意味する。Huでは「－」の代わりに「，」を使うこともできる。ラベルが使えるBASICでは，ここでも原則としてラベルが使える。
式	0とか5と言った定数や変数，関数等。また，それらを組み合わせたもの。
変数名	配列変数名（添字〔，添字…〕）を含む。文字型でも数値型でも良い。
“文字列”	文字型の変数や式でも良い。“文字型データ”や“ファイル名”でも同様。
日本語や小文字	カナや漢字，アルファベットの小文字，〔 〕，｛ ｝は説明用なので実際には指定しない（他のキーボードにない記号も同様)。ただし，アルファベットの大文字，数字，カンマなどキーボードにある記号は実際にも入力する。
［ ］	この中を省略しても良いことを示す。〔A〔B〔C〕〕〕なら全部省略，A，AB，ABCが指定できる。ただしBCのような指定はできない。
…	同様のパラメータが続くことを示す。
{○ {△	○か△のどちらか(3個以上ではどれか)を選ぶことを示す。また，[{○/△}]とあれば，○か△か省略のどれかになる。
(Xs, Ys)－(Xe, Ye)	（左上のx座標，左上のy座標）－（右下のx座標，右下のy座標）
(x，y)	N88，MSX，M25ではSTEP(x，y)としても良い。x，y以外にx1，y2等も同様(xs，ye等は除く)。
S25のx，y	同様にSTEP(x，y)としても良い。( )が付くので注意。
【カラー指定】	〔[〔色〕〔，機能〕〔，ラインスタイル〕]〕…S25 カラー指定1としたもののみ 〔[色〕〔，重ね合わせ〕]〕…S 〔[色〕〔，W重ね合わせ〕]〕…SHARPの④⑥ 指定しない(指定できない)…SHARPの⓪①②③⑤ （[ ]の中は無視してください)
文，関	本表はステートメント・コマンドと関数とを完全に分けているが，例外として文はステートメント，関は関数である。

表2 各BASICのグラフィック命令パラメータ

モード	グラフィックの各プレーンのドットの様子	指定したパラメータ	結果	機能
PSET ・シャープ系の機能＝0	0 0 1 1	0 1 0 1	0 1 0 1	強制的に指定の色に変える。
PRESET	0 0 1 1	0 1 0 1	1 0 1 0 または 0 0 0 0 または 1 1 1 1	強制的に指定の色の補色か，背景色か，0(Hu)にする。
AND ・S25の機能＝3	0 0 1 1	0 1 0 1	0 0 0 1	両方セットされていればセット。
OR ・シャープ系の機能＝1	0 0 1 1	0 1 0 1	0 1 1 1	どちらかがセットされていればセット。
XOR ・S25の機能＝2	0 0 1 1	0 1 0 1	0 1 1 0	反転させる。
NOT	0 0 1 1	0 1 0 1	1 0 1 0	強制的に補色でセット。 (F用)
SCR	0 0 1 1 0 0 1 1	0 0 1 0 0 0 1 0 0 1 1 1 0 1 1 1	0 0 1 0 0 1 1 1	パラメータは前者が色，後者がパターン。色が1の所のみパターンをセット。 (M25用)
FPSET	0 0 1 1 0 0 1 1	0 0 1 0 0 0 1 0 0 1 1 1 0 1 1 1	0 0 1 1 0 1 0 1	パラメータは前者が色，後者がパターン。パターンが1の所のみ色をセット。 (M25用)

表3 MZ-80B/2000/2200で表中にない命令

表中では割愛したMZ-80B/2000/2200用BASIC MZ-2Z001，MZ-2Z002，MZ-2Z021でサポートされているGPIBインタフェイス(オプション)サポート命令。

ICL	REN	LCL	LLO	DCL
TRG	PCT	WRT	RED	WRT/
RED/	CMDW	CMDR	ON SRQ	SPOL
PPC	PPOL	PPU	GPIBM	EOIW
EOIR				

いずれも他のBASICには移植不能。

表4 MZ-700/1500で表中にない命令

表中では割愛したMZ-700のHuBASIC，S-BASIC及びMZ-1500のS-BASICでサポートされているプロッタプリンタ命令(( )内はMZ-1500での名称)。

MODE TN (PMODE GR)	MODE TL (PMODE TL)	MODE TS (PMODE TS)
MODE GR (PMODE GR)	PCOLOR	TEST (PTEST)
SKIP (PSKIP)	PAGE	LINE (PLINE)
RLINE	MOVE (PMOVE)	RMOVE (PMOV)
PHOME	HSET	GPRINT
AXIS	CIRCLE (PCIRCLE)	PLOT ON
PLOT OFF		

いずれも他のBASICには移植不能(プロッタプリンタを使えばPRINT/P・LPRINTで代用できる場合もある)。

1.1 RUN CONT NEW NEW ON CMD UNLINK TRON TROFF MON LIST LLIST AUTO EDIT

分類	働き	N-BASIC (PC-8001ほか) Ⓓ…ディスク版にのみ存在	N88BASIC (PC-8801ほか) ⓈⓇ…88SRのサウンド関係の拡張命令 Ⓓ…ディスク版にのみ存在	MSX-BASIC V1.0 (各社MSX) Ⓓ…ディスク版にのみ存在
一般コマンド	プログラムを先頭から実行。	RUN	RUN	RUN
	変数を消してプログラムを途中から実行。	RUN 行番号	RUN 行番号	RUN 行番号
	指定プログラムをロードして先頭から実行。	RUN "ファイル名" Ⓓ	RUN "ファイル名" Ⓓ	RUN "ファイル名"
	中断したプログラムを再開。	CONT	CONT	CONT
	指定の位置でプログラムを中断する。			
	中断したプログラムの所から指定ステップ実行。			
	プログラムを全部抹消。	NEW	NEW	NEW
	BASICのモードを切り替える。		NEW ON ディップスイッチの状態 CMD UNLINK ⓈⓇ	
	BASICプログラムの格納開始アドレスを変更。			
	トレースモードに入る(実行中の行番号を画面に表示)。	TRON	TRON	TRON
	トレースモードに入る(行番号はプリンタに出力)。			
	トレースモードから抜ける。	TROFF	TROFF	TROFF
	モニタに入る。	MON	MON	
	IPLを起動する。		(NEW ON モード)	
	リストを画面に表示。	LIST 〔範囲〕	LIST 〔範囲〕	LIST 〔範囲〕
	プリンタにリストを出力。	LLIST 〔範囲〕	LLIST 〔範囲〕	LLIST 〔範囲〕
	指定ファイル(デバイス)にリストを出力。			
	'の後だけリストを画面に表示。			
	'の後だけリストをプリンタに出力。			
	'の後だけリストをファイル(デバイス)に出力。			
	リスト出力時の範囲のデフォルトを指定する。			
	行番号を自動発生する。	AUTO 〔開始行〕〔,増分〕	AUTO 〔開始行〕〔,増分〕	AUTO 〔開始行〕〔,増分〕
	'付きの行番号を自動発生する。			
	指定行をエディットする。		EDIT 行番号 Ⓓ	
	エディットモードに入る。		EDIT 行番号 Ⓓ	

86ページに続く

注　このDATA LISTは4ページ単位で構成されています。偶数ページ欄外は見開き2ページで説明されているコマンド，ステートメントです。比較したBASICは，N-BASIC, N88BASIC，MSX-BASIC V1.0, F-BASIC V3.0, HuBASIC(MZ-700/X1), BASIC-M25/S25, S-BASIC, SHARP BASICです。

F-BASIC V3.0（FM-7ほか） Ⓓ…ディスク版にのみ存在	HuBASIC（MZ-700）	HuBASIC(X1・X1 turbo) ①CZ-8CB01 V1.0　③CZ-8CB01 V2.0，CZ-8FB01 V2.0 ②CZ-8FB01 V1.0　④CZ-8FB02 V1.0にのみ存在
RUN	RUN	RUN
RUN 行番号	RUN 行番号	RUN 行番号
RUN "ファイル名"	RUN "ファイル名"	RUN "ファイル名"
CONT	CONT	CONT
NEW	NEW	NEW
		NEW ON モード ③④
	NEW ON アドレス	NEW ON アドレス
TRON	TRON	TRON
TROFF	TROFF	TROFF
MON	MON	MON
	BOOT	BOOT
LIST〔範囲〕	LIST〔範囲〕	LIST〔範囲〕
LLIST〔範囲〕	LLIST〔範囲〕	LLIST〔範囲〕
LIST"ファイル名"〔,範囲〕	LIST "ファイル名"〔,範囲〕	LIST "ファイル名"〔,範囲〕
		LIST＊〔範囲〕④
		LLIST＊〔範囲〕④
		LIST＊ "ファイル名"〔,範囲〕④
		LIST@ 〔範囲〕④
AUTO〔開始行〕〔,増分〕	AUTO〔開始行〕〔,増分〕	AUTO〔開始行〕〔,増分〕
		AUTO＊〔開始行〕〔,増分〕④
EDIT 行番号	EDIT〔行番号〕	EDIT〔行番号〕
		EDIT〔行番号〕④

87ページに続く

1.2 RUN CLR:RUN CONT STOP STEP NEW NEW ON TRON TRLON TRON/P TROFF MON BYE BOOT LIST LLIST LIST/P AUTO EDIT

BASIC-M25 (MZ-2500)	BASIC-S25(MZ-2500)	S-BASIC(MZ-700/1500) ①MZ-700 テープ版(付属) ②MZ-700 QD・Disk版 ③MZ-1500 QD・Disk版 にのみ存在
RUN	RUN	RUN
RUN 行番号	CLR:RUN 行番号	CLR:RUN 行番号
RUN "ファイル名"	RUN "ファイル名"	RUN "ファイル名"
CONT	CONT	CONT
STOP 行番号	STOP 行番号	
STEP 〔行数〕	STEP 〔行数〕	
NEW	NEW	NEW
NEW ON モード	NEW ON モード	NEW ON モード ③
TRON 〔範囲〕	TRON 〔範囲〕	TRON ③
LTRON 〔範囲〕	LTRON 〔範囲〕	TRON/P ③
TROFF	TROFF	TROFF ③
MON	MON	BYE
BOOT	BOOT	BOOT
LIST 〔範囲〕	LIST 〔範囲〕	LIST 〔範囲〕
LLIST 〔範囲〕	LIST/P 〔範囲〕	LIST/P 〔範囲〕
LIST "ファイル名"〔,範囲〕	LIST "ファイル名"〔,範囲〕	
LIST＊〔範囲〕	LIST＊〔範囲〕	
LLIST＊〔範囲〕	LLIST＊〔範囲〕	
LIST＊ "ファイル名"〔,範囲〕	LIST＊ "ファイル名"〔,範囲〕	
AUTO 〔開始行〕〔,増分〕	AUTO 〔開始行〕〔,増分〕	AUTO 〔開始行〕〔,増分〕
AUTO＊	AUTO＊	
EDIT 〔行番号〕	EDIT 〔行番号〕	
EDIT 〔行番号〕	EDIT 〔行番号〕	

88ページに続く

SHARP BASIC (MZ-80K/C/1200/80B/2000/2200) ⓪SP-5030 ③MZ-1Z001 ⑤MZ-2Z001 ①SB-5520 ④MZ-1Z002 ⑥MZ-2Z021 ②SB-6520 にのみ存在	備考	働き	分類
RUN		プログラムを先頭から実行。	一般コマンド
CLR:RUN 行番号	シャープ系のRUN 行番号はGOTO 行番号と同じように変数をクリアしないので同じ意味にするにはCLRを付ける必要がある。	変数を消してプログラムを途中から実行。	
RUN “ファイル名” ②⑤⑥	同じような働きをするものに，LOAD “ファイル名”, R と CHAIN “ファイル名” がある。	指定プログラムをロードして先頭から実行。	
CONT	プログラム中で使われることはない。	中断したプログラムを再開。	
	プログラム中で使われることはない。	指定の位置でプログラムを中断する。	
	プログラム中で使われることはない。	中断したプログラムの所から指定ステップ実行。	
NEW		プログラムを全部抹消。	
	N88のNEW ONはN88 BASICとN-BASIC，画面モード等を変えるのに対し，その他ではBASIC等の一部機能をカットする(CMD UNLINKはサウンド用のCMD文をカットする)。	BASICのモードを切り替える。	
	マシン語関連(移植困難)。ただしプログラム中で使われることはない。	BASIC プログラムの格納開始アドレスを変更。	
	プログラム中で使っても意味がないので，出てこない。	トレースモードに入る(実行中の行番号を画面に表示)。	
	プログラム中で使っても意味がないので，出てこない。	トレースモードに入る(行番号はプリンタに出力)。	
	プログラム中で使っても意味がないので，出てこない。	トレースモードから抜ける。	
{BYE ⓪ {MON①～⑥}		モニタに入る。	
BOOT ①～⑥		IPLを起動する。	
LIST〔範囲〕	HuBASIC等はリスト出力後プログラムが止まらない（マイクロソフトでは止まる）ので，プログラム中に出てくることもある。	リストを画面に表示。	
LIST／P〔範囲〕	HuBASIC等はリスト出力後プログラムが止まらない（マイクロソフトでは止まる）ので，プログラム中に出てくることもある。	プリンタにリストを出力。	
	範囲を省略するとSAVE “ファイル指定”, Aと同意になる。	指定ファイル（デバイス）にリストを出力。	
		'の後だけリストを画面に表示。	
	簡易ワープロのように使える命令。メッセージ出力用にPRINT等の代わりに使うことがあるかもしれないが，これがないBASICでは本来のPRINT文等に戻す。	'の後だけリストをプリンタに出力。	
		'の後だけリストをファイル(デバイス)に出力。	
	主にコマンドの使いやすさを図るためのものなので，プログラム中には出てこない。	リスト出力時の範囲のデフォルトを指定する。	
AUTO〔開始行〕〔, 増分〕①～⑥	プログラム中で使われることはない。	行番号を自動発生する。	
	プログラム中で使われることはない。	'付きの行番号を自動発生する。	
	プログラム中で使われることはない。	指定行をエディットする。	
	プログラム中で使われることはない。	エディットモードに入る。	

89ページに続く

2.1 SEARCH DELETE RENUM TERM ASK FILES LFILES CLOAD LOAD CSAVE SAVE CLOAD? LOAD? SKIPF VERIFY CHAIN COMMON MERGE BLOAD LOADM BSAVE SAVEM

分類	働き	N-BASIC（PC-8001ほか） Ⓓ…ディスク版にのみ存在	N88BASIC（PC-8801ほか） ⓈⓇ…88SRのサウンド関係の拡張命令 Ⓓ…ディスク版にのみ存在	MSX-BASIC V1.0（各社MSX） Ⓓ…ディスク版にのみ存在
一般コマンド	プログラム中から文字列を捜し出し，あった行を画面に表示。			
	プログラム中から文字列を捜し，プリンタにその行を出力。			
	プログラムの一部分を削除する。	DELETE 範囲	DELETE 範囲	DELETE 範囲
	行番号を付けなおす。	RENUM〔新行番号〕〔,〔旧行番号〕〔,,増分〕〕	RENUM〔新行番号〕〔,〔旧行番号〕〔,増分〕〕	RENUM〔新行番号〕〔,〔旧行番号〕〔,増分〕〕
	ターミナルモードに入る。	TERM パラメータ	TERM "パラメータ"〔,〔モード〕〔,変数領域の大きさ〕〕	
	予約語の大文字・小文字表示を切り替える。			
	対話型のテレビタイマ予約を行う。			
ファイル操作コマンド（ステートメント）	登録されているファイルを表示。	FILES〔ドライブ番号〕Ⓓ	FILES〔ドライブ番号〕Ⓓ	FILES〔"デバイス名"〕Ⓓ
	登録されているファイルをプリンタに出力。	LFILES〔ドライブ番号〕Ⓓ	LFILES〔ドライブ番号〕Ⓓ	LFILES〔"デバイス名"〕Ⓓ
	BASICのプログラムをカセットからロード。	CLOAD "ファイル名"	LOAD "CAS:ファイル名"	CLOAD〔"ファイル名"〕
	BASICのプログラムをロード。	LOAD "ファイル名"〔,R〕Ⓓ	LOAD "ファイル名"〔,R〕	LOAD "ファイル名"〔,R〕
	BASICのプログラムをカセットにセーブ。	CSAVE "ファイル名"	SAVE "CAS:ファイル名"	CSAVE "ファイル名"〔,ボーレイト〕
	BASICのプログラムをセーブ。	SAVE "ファイル名"〔,A〕Ⓓ	SAVE "ファイル名"〔,{A/P}〕	SAVE "ファイル名"〔,A〕
	プログラムをベリファイ。	CLOAD? "ファイル名"	LOAD? "CAS:ファイル名"	CLOAD?〔"ファイル名"〕
	変数を保護しつつプログラムをロードして実行（プログラム単位のGOTO）。		CHAIN〔MERGE〕"ファイル名"〔,実行開始行番号〕〔,ALL〕〔,DELETE 範囲〕Ⓓ	
	CHAINで受け渡す（クリアしない）変数を指定する。		COMMON 変数名〔,変数名…〕Ⓓ	
	プログラムを一時的に呼び出して実行（プログラム単位のGOSUB）。			
	プログラムを併合する。	MERGE "ファイル名" Ⓓ	MERGE "ファイル名" Ⓓ	MERGE "ファイル名"
	マシン語プログラムをロードする。		BLOAD "ファイル名"〔,〔ロードアドレス〕〔,R〕〕Ⓓ	BLOAD "ファイル名"〔,R〕〔,オフセットアドレス〕
	マシン語プログラムをセーブする。		BSAVE "ファイル名",セーブ開始アドレス,長さ Ⓓ	BSAVE "ファイル名",セーブ開始アドレス,セーブ終了アドレス〔,実行開始アドレス〕
	グラフィック画面の内容をロードする。			

90ページに続く

F-BASIC V3.0 (FM-7ほか) Ⓓ…ディスク版にのみ存在	HuBASIC (MZ-700)	Hu BASIC (X1・X1 turbo) ①CZ-8CB01 V1.0 ③CZ-8CB01 V2.0, CZ-8FB01 V2.0 ②CZ-8FB01 V1.0 ④CZ-8FB02 V1.0にのみ存在
	SEARCH "文字列"	SEARCH "文字列"
DELETE 範囲	DELETE 範囲	DELETE 範囲
RENUM〔新行番号〕〔,〔旧行番号〕〔,増分〕〕	RENUM〔新行番号〕〔,〔旧行番号〕〔,増分〕〕	RENUM〔新行番号〕〔,〔旧行番号〕〔,増分〕〕
TERM "パラメータ"		
		ASK
FILES〔"デバイス名"〕	FILES〔"デバイス名"〕	FILES〔"デバイス名"〕
FILES〔"デバイス名"〕〔,L〕	LFILES〔"デバイス名"〕	LFILES〔"デバイス名〕
LOAD〔"CAS:ファイル名"〕	LOAD〔"ファイル名"〕	LOAD〔"CAS:ファイル名"〕
LOAD〔"ファイル名"〕〔,R〕	LOAD〔"ファイル名"〕	LOAD〔"ファイル名"〕
SAVE "CAS:ファイル名"	SAVE〔"ファイル名"〕	SAVE〔"CAS:ファイル名"〕
SAVE "ファイル名"〔,A〕〔,P〕	SAVE〔"ファイル名"〕〔,A〕	SAVE〔"ファイル名"〕〔,A〕
LOAD?〔"ファイル名"〕 またはSKIPF〔"ファイル名"〕	LOAD?〔"ファイル名"〕 またはVERIFY〔"ファイル名"〕	LOAD?〔"ファイル名"〕 またはVERIFY〔"ファイル名"〕
CHAIN〔MERGE〕"ファイル名"〔,実行開始行番号〕〔,ALL〕〔,DELETE 範囲〕 Ⓓ		CHAIN "ファイル名"
COMMON 変数名〔,変数名…〕 Ⓓ		(自動的に全変数が受け渡される)
		CHAINで代用
MERGE〔"ファイル名"〕	MERGE〔"ファイル名"〕	MERGE〔"ファイル名"〕
LOADM"ファイル名"〔,〔オフセットアドレス〕〔,R〕〕	LOADM"ファイル名"〔,〔ロードアドレス〕〔,R〕〕	LOADM"ファイル名"〔,〔ロードアドレス〕〔,R〕〕
SAVEM"ファイル名",セーブ開始アドレス,セーブ終了アドレス,実行開始アドレス	SAVEM"ファイル名",セーブ開始アドレス,セーブ終了アドレス〔,実行開始アドレス〕	SAVEM"ファイル名",セーブ開始アドレス,セーブ終了アドレス〔,実行開始アドレス〕

91ページに続く

2.2 SEARCH LSERCH SEARCH/P DELETE RENUM TERM OPTIONLIST FILES DIR LFILES DIR/P LOAD SAVE VERIFY CHAIN COMMON SWAP MERGE APPEND BLOAD BSAVE GLOAD

BASIC-M25 (MZ-2500)	BASIC-S25(MZ-2500)	S-BASIC(MZ-700/1500) ①MZ-700 テープ版(付属) ②MZ-700 QD・Disk版 ③MZ-1500QD・Disk版 にのみ存在
SEARCH "文字列"〔,範囲〕	SEARCH"文字列"〔,範囲〕	SEARCH "文字列"
LSEARCH "文字列" 〔,範囲〕	SEARCH/P"文字列"〔,範囲〕	
DELETE 範囲	DELETE 範囲	DELETE 範囲
RENUM 〔新行番号〕〔,〔増分〕〔,範囲〕〕	RENUM 〔新行番号〕〔,〔増分〕〔,範囲〕〕	RENUM 〔新行番号〕〔,〔旧行番号〕〔,増分〕〕
TERM "パラメータ"	TERM "パラメータ"	
OPTION LIST {UCASE / LCASE}	OPTION LIST {UCASE / LCASE}	
FILES 〔"デバイス名"〕	DIR 〔"デバイス名"〕	DIR〔デバイス名〕またはDIR 〔"デバイス名"〕
LFILES 〔"デバイス名"〕	DIR/P 〔"デバイス名"〕	DIR/P 〔デバイス名〕またはDIR/P 〔"デバイス名"〕
LOAD "CMT:ファイル名"	LOAD "CMT:ファイル名"	LOAD 〔"CMT:ファイル名"〕〔,A②③〕
LOAD "ファイル名"	LOAD "ファイル名"	LOAD 〔"ファイル名"〕
		SAVE "CMT:ファイル名"
SAVE "ファイル名"〔,A〕	SAVE "ファイル名"〔,A〕	SAVE "ファイル名〔,A ②③〕
		VERIFY〔"ファイル名"〕
CHAIN〔MERGE〕"ファイル名"〔,実行開始行番号〕〔,ALL〕〔,DELETE 範囲〕	CHAIN 〔MERGE〕 "ファイル名" 〔,実行開始行番号〕〔,ALL〕 〔,DELETE 範囲〕	CHAIN "ファイル名" ②③
COMMON 変数名〔,変数名…〕	COMMON 変数名〔,変数名…〕	(自動的に全変数が受け渡される ②③)
CHAINで代用	SWAP "ファイル名"	SWAP "ファイル名" ②③
MERGE "ファイル名"	MERGE "ファイル名"	MERGE "ファイル名"〔,A〕
BLOAD"ファイル名"〔,ロードアドレス〕	LOAD "ファイル名"〔,ロードアドレス〕	LOAD "ファイル名"
BSAVE "ファイル名",セーブ開始アドレス,長さ〔,〔実行開始アドレス〕〔,ロード時アドレス〕〕	SAVE"ファイル名",セーブ開始アドレス,長さ〔,〔実行開始アドレス〕〔,ロード時アドレス〕〕	(モニタで行う)
GLOAD "ファイル名"〔,(x_s, y_s)〕	GLOAD "ファイル名"〔,x_s, y_s〕	

92ページに続く

SHARP BASIC (MZ-80K/C/1200/80B/2000/2200) ⓪ SP-5030 ③ MZ-1Z001 ⑤ MZ-2Z001 ① SB-5520 ④ MZ-1Z002 ⑥ MZ-2Z021 ② SB-6520 にのみ存在	備考	働き	分類
SEARCH "文字列"〔, 範囲〕⑥	プログラム中で使われることはない。	プログラム中から文字列を捜し出し，あった行を画面に表示。	一般コマンド
SEARCH/P "文字列"〔, 範囲〕⑥	プログラム中で使われることはない。	プログラム中から文字列を捜し，プリンタにその行を出力。	一般コマンド
DELETE 範囲 ⑥	まれにHu BASIC等ではPCGのセットのように一度しか使わない部分をこれでカットし，あらかじめKEY 0文でセットしておいた"GOTO"等でプログラムの実行を再開することがある(変数やマシン語のエリアが足りないときに出て来ることがある)。	プログラムの一部分を削除する。	一般コマンド
RENUM〔新行番号〕〔,〔増分〕〔,範囲〕〕⑥	プログラム中で使われることはない。	行番号を付けなおす。	一般コマンド
	プログラム中で使われることはない。	ターミナルモードに入る。	一般コマンド
	プログラム中で使われることはない。	予約語の大文字・小文字表示を切り替える。	一般コマンド
		対話型のテレビタイマ予約を行う。	一般コマンド
DIR〔FDn〕(nは1～4のドライブ番号) ②⑤⑥	ファイルの名を入力するとき，参考のためにプログラム中で使われることもある(SHARP BASICでは取ってしまうこと)。	登録されているファイルを表示。	ファイル操作コマンド(ステートメント)
DIR/P〔FDn〕(nは1～4のドライブ番号) ②⑤⑥		登録されているファイルをプリンタに出力。	ファイル操作コマンド(ステートメント)
LOAD/T〔"ファイル名"〕①～⑥	,Rを付けるとそのまま実行に移るので，プログラム中で使われることがある。その場合はCHAIN文に変更する。CAS:やCMT:，/Tはテープ版では省略可。AはS-BASIC特有で，SAVE～,A でセーブしたものをロードするもの。	BASICのプログラムをカセットからロード。	ファイル操作コマンド(ステートメント)
LOAD "ファイル名"		BASICのプログラムをロード。	ファイル操作コマンド(ステートメント)
SAVE/T "ファイル名" ①～⑥	MSXのボーレイトは1で1200ボー，2で600ボー。	BASICのプログラムをカセットにセーブ。	ファイル操作コマンド(ステートメント)
SAVE "ファイル名"	,Aを付けるとアスキーセーブする。,Pを付けるとプロテクトをかけてセーブする(それをロードするとリスト，セーブできない)。	BASICのプログラムをセーブ。	ファイル操作コマンド(ステートメント)
VERIFY〔"ファイル名"〕		プログラムをベリファイ。	ファイル操作コマンド(ステートメント)
CHAIN "ファイル名" ②⑤⑥	ALL指定のできるBASICでは，ALLを付けると全変数が受け渡されるが，省略するとCOMMON文で指定した変数のみが受け渡される。その場合COMMONがなければ全変数がクリアされるので，ALL指定のできない(自動的に全変数が受け渡される)BASICでは，CHAINでなくRUN "ファイル名"を使っても良い。また実行開始行番号が指定されているプログラムをそれができないBASICに移植する場合は，CHAIN前にXX=999のようにしておき，CHAINした先のプログラムでIF XX=999 THEN 1000のようにする。つまり変数を途中からスタートさせるためのフラグにする(CLEARかLIMITで確保したマシン語エリアでPOKE・PEEKしても良い)。	変数を保護しつつプログラムをロードして実行(プログラム単位のGOTO)。	ファイル操作コマンド(ステートメント)
(自動的に全変数が受け渡される ②⑤⑥)	受け渡す変数を限定して，メモリのフリーエリアを少しでも広くするために使われる。	CHAINで受け渡す(クリアしない)変数を指定する。	ファイル操作コマンド(ステートメント)
SWAP "ファイル名" ②⑤⑥	CHAINで代用する方法：SWAPそのものはCHAIN(M25ではALLを付ける)に置き換える。SWAPした先ではENDかプログラムの末尾でもとのプログラムに戻るので，もとのプログラムの途中から実行するようなCHAIN文を置く(X 1/turbo ではCHAINの項参照)。	プログラムを一時的に呼び出して実行(プログラム単位のGOSUB)。	ファイル操作コマンド(ステートメント)
APPEND "ファイル名" ②⑤⑥	S-BASICの ,Aはアスキーセーブされたものが対象であることを示す。	プログラムを併合する。	ファイル操作コマンド(ステートメント)
LOAD "ファイル名"	マシン語プログラムは，ロードすることができても移植にはマシン語の知識が必要。	マシン語プログラムをロードする。	ファイル操作コマンド(ステートメント)
(モニタで行う)		マシン語プログラムをセーブする。	ファイル操作コマンド(ステートメント)
		グラフィック画面の内容をロードする。	ファイル操作コマンド(ステートメント)

93ページに続く

分類	働き	N-BASIC (PC-8001ほか) Ⓓ…ディスク版にのみ存在	N88BASIC (PC-8801ほか) SR…88SRのサウンド関係の拡張命令 Ⓓ…ディスク版にのみ存在	MSX-BASIC V1.0 (各社MSX) Ⓓ…ディスク版にのみ存在
ファイル操作コマンド(ステートメント)	グラフィック画面の内容をセーブする。			
	クイックディスクの全内容を64KRAMカードにロードする。			
	64KRAMカードの全内容をクイックディスクにセーブする。			
	ディスクの使用を始める。	MOUNT〔ドライブ番号〕Ⓓ	(不要)	(不要)
	ディスクの使用を終了する。	REMOVE〔ドライブ番号〕Ⓓ	(不要)	(不要)
一般ステートメント	変数に値を代入する。	〔LET〕 変数名 ＝ 値	〔LET〕 変数名 ＝ 値	〔LET〕 変数名 ＝ 値
	指定の英字から始まる変数のデフォルトの型を整数にする。	DEFINT 英字〔-英字〕〔,…〕	DEFINT 英字〔-英字〕〔,…〕	DEFINT 英字〔-英字〕〔,…〕
	指定の英字から始まる変数のデフォルトの型を単精度にする。	DEFSNG 英字〔-英字〕〔,…〕	DEFSNG 英字〔-英字〕〔,…〕	DEFSNG 英字〔-英字〕〔,…〕
	指定の英字から始まる変数のデフォルトの型を倍精度にする。	DEFDBL 英字〔-英字〕〔,…〕	DEFDBL 英字〔-英字〕〔,…〕	DEFDBL 英字〔-英字〕〔,…〕
	指定の英字から始まる変数のデフォルトの型を文字型にする。	DEFSTR 英字〔-英字〕〔,…〕	DEFSTR 英字〔-英字〕〔,…〕	DEFSTR 英字〔-英字〕〔,…〕
	ユーザー関数を定義する。	DEF FN関数名〔(仮引数〔,…〕)〕＝式	DEF FN関数名〔(仮引数〔,…〕)〕＝式	DEF FN関数名〔(仮引数〔,…〕)〕＝式
	配列を宣言する。	DIM 配列名(添字の最大値〔,…〕)〔,配列名…〕	DIM 配列名(添字の最大値〔,…〕)〔,配列名…〕	DIM 配列名(添字の最大値〔,…〕)〔,配列名…〕
	配列の下限の添字を0か1にする。		OPTION BASE {0 / 1}	
	配列を消去する。	ERASE 配列名〔,配列名…〕	ERASE 配列名〔,配列名…〕	ERASE 配列名〔,配列名…〕
	角度の単位を変更する。			
	注釈を置く(ステートメントに続けるには前に:が必要)。	REM〔注釈〕	REM〔注釈〕	REM〔注釈〕
	注釈を置く(ステートメントに続けるときも:は不要)。	'〔注釈〕	〔注釈〕	'〔注釈〕
	乱数の発生系列を変更する。		RANDOMIZE〔系列〕	

94ページに続く

F-BASIC V3.0 (FM-7ほか) Ⓓ…ディスク版にのみ存在	HuBASIC (MZ-700)	HuBASIC(X1・X1 turbo) ①CZ-8CB01 V1.0 ③CZ-8CB01 V2.0, CZ-8FB01 V2.0 ②CZ-8FB01 V1.0 ④CZ-8FB02 V1.0にのみ存在
(不要)		(不要)
(不要)		(不要)
〔LET〕 変数名 = 値	〔LET〕 変数名 = 値	〔LET〕 変数名 = 値
DEFINT 英字〔-英字〕〔,…〕	DEFINT 英字〔-英字〕〔,…〕	DEFINT 英字〔-英字〕〔,…〕
DEFSNG 英字〔-英字〕〔,…〕	DEFSNG 英字〔-英字〕〔,…〕	DEFSNG 英字〔-英字〕〔,…〕
DEFDBL 英字〔-英字〕〔,…〕	DEFDBL 英字〔-英字〕〔,…〕	DEFDBL 英字〔-英字〕〔,…〕
DEFSTR 英字〔-英字〕〔,…〕	DEFSTR 英字〔-英字〕〔,…〕	DEFSTR 英字〔-英字〕〔,…〕
DEF FN関数名〔(仮引数〔,…〕)〕=式	DEF FN関数名〔(仮引数〔,…〕)〕= 式	DEF FN関数名〔(仮引数〔,…〕)〕=式
DIM 配列名(添字の最大値〔,…〕)〔,配列名…〕	DIM 配列名(添字の最大値〔,…〕)〔,配列名…〕	DIM 配列名(添字の最大値〔,…〕)〔,配列名…〕
	OPTION BASE {0 / 1}	OPTION BASE {0 / 1}
ERASE 配列名〔,配列名…〕		ERASE 配列名〔,配列名…〕 ④
REM 〔注釈〕	REM 〔注釈〕	REM 〔注釈〕
'〔注釈〕	'〔注釈〕	'〔注釈〕
RANDOMIZE 〔系列〕		RANDOMIZE 〔系列〕

95ページに続く

3.2 GSAVE LOAD ALL SAVE ALL LET DEF INT DEF SNG DEF DBL DEF STR DEF FN DIM OPTION BASE ERASE OPTION ANGLE REM ' RANDOMIZE

BASIC-M25 (MZ-2500)	BASIC-S25(MZ-2500)	S-BASIC(MZ-700/1500) ① MZ-700 テープ版(付属) ② MZ-700 QD・Disk版 ③ MZ-1500QD・Disk版 にのみ存在
GSAVE "ファイル名"〔, (x_s, y_s) − (x_e, y_e)〕	GSAVE "ファイル名"〔, x_s, y_s, x_e, y_e〕	
		LOAD ALL ③
		SAVE ALL ③
(不要)	(不要)	(不要)
(不要)	(不要)	(不要)
〔LET〕 変数名 = 値	〔LET〕 変数名 = 値	〔LET〕 変数名 = 値
DEF INT 英字〔-英字〕〔,…〕	DEF INT 英字〔-英字〕〔, …〕	
DEF SNG 英字〔-英字〕〔,…〕	DEF SNG 英字〔-英字〕〔, …〕	
DEF DBL 英字〔-英字〕〔,…〕	DEF DBL 英字〔-英字〕〔, …〕	
DEF STR 英字〔-英字〕〔,…〕	DEF STR 英字〔-英字〕〔, …〕	
DEF FN関数名〔(仮引数〔,…〕)〕=式	DEF FN関数名〔(仮引数〔,…〕)〕=式	DEF FN関数名〔(仮引数〔,…〕)〕=式
DIM配列名(添字の最大値〔,…〕)〔,配列名…〕	DIM配列名(添字の最大値〔,…〕)〔,配列名…〕	DIM 配列名(添字の最大値〔,…〕)〔,配列名…〕
OPTION BASE {0 / 1}	OPTION BASE {0 / 1}	
ERASE 配列名〔,配列名…〕	ERASE 配列名〔,配列名…〕	
OPTION ANGLE {RADIANS / DEGREES}	OPTION ANGLE {RADIANS / DEGREES}	
REM 〔注釈〕	REM 〔注釈〕	REM 〔注釈〕
' 〔注釈〕	' 〔注釈〕	' 〔注釈〕
RANDOMIZE 〔系列〕	RANDOMIZE 〔系列〕	

96ページに続く

SHARP BASIC (MZ-80K/C/1200/80B/2000/2200) ⓪ SP-5030　③ MZ-1Z001　⑤ MZ-2Z001 ① SB-5520　④ MZ-1Z002　⑥ MZ-2Z021 ② SB-6520　にのみ存在	備　考	働　き	分類
		グラフィック画面の内容をセーブする。	ファイル操作コマンド(ステートメント)
		クイックディスクの全内容を64KRAMカードにロードする。	
		64KRAMカードの全内容をクイックディスクにセーブする。	
(不要)	ディスケットを入れたときと出す前とに必要なもので，FATの読み書きを行うもの。単純に取ってしまって良い。	ディスクの使用を始める。	
(不要)		ディスクの使用を終了する。	
〔LET〕変数名 ＝ 値		変数に値を代入する。	一般ステートメント
	S-BASIC，SHARP BASICでは，DEFINT 等はないので，取ってしまう。ただし，DEFSTRが使われていた場合は，以後の変数を変更する必要がある。例えば DEFSTR A-C，K，S-V となっていたら，A，B，C，K，S，T，U，V から始まる型指定のない変数（%！#$ の付いていない変数）は $を付ける。%！#は通常取ることになるので，例えば AX%，BCD，D！，GR#，H0，J$ という変数は，AX，BCD$，D，GR，H0，J$ に直すことになる。 なお，CLEAR，RUN 時には，MSX以外ではDEFSNG A-Zが，MSXではDEFDBLA-Zが自動的に実行される。ただしturboとMZ-2500の 漢字変数名では%等の記号がなければ常に倍精度。	指定の英字から始まる変数のデフォルトの型を整数にする。	
		指定の英字から始まる変数のデフォルトの型を単精度にする。	
		指定の英字から始まる変数のデフォルトの型を倍精度にする。	
		指定の英字から始まる変数のデフォルトの型を文字型にする。	
DEF FN関数名(仮引数)＝式	SHARP BASICへの移植では，引数・結果共に数値型，引数は1個，関数名は1文字と，仕様が厳しく限定されているため，場合によってはFNを実際に使っている所で展開する必要がある。例えば，DEF FNABC$(A$, B, C)＝LEFT$(RIGHT$(A$, B), (,)＋B$で，PRINT FNABC$(X$, 5, 2)なら，DEF FNは省略し，PRINT LEFT$ (RIGHT$ (X$, 5), 2)＋B$にする。FNが使われている所全部でこれを行う。	ユーザー関数を定義する。	
DIM 配列名(添字の最大値〔,…〕)〔,配列名…〕	SHARP BASICへの移植では，2次元までしか許されない点に注意する。また，添字が 255 までしか許されないため，場合によっては次元の変換を行う。例えばDIM A$(1999)でPRINT A$(X)とあれば，DIM A$(255, 7)でPRINT A$ (X／8，X-INT (X／8) ＊8) とする(代入時も添字の変換を行う)。またDIM B(5，2，8)でPRINT B(I，J，K) なら，DIM B(161)で，PRINT B (I＋6＊(J＋3＊K))にする。BASIC 内部では何次元であっても結局は1次元に変換されるので，原理上，次元の変換は可能なことである。	配列を宣言する。	
	配列の添字が1以上しか使わないときは，0の分を確保しないことでメモリが少しだけ節約できるようOPTION BASE 1としているが，単に取ってしまって良い。	配列の下限の添字を0か1にする。	
	DIMと反対の働きをするステートメント。ERASEの主な使い方は3つある。1つは不要になった配列を消して，フリーエリアを増やすこと。この場合はメモリが足りなくなる可能性はあるが，単にERASE文を取ってしまう。次に，1度ERASE して，次に前回と同じ条件（大きさ・次元）のDIMを行う場合。この時は配列の各要素を0または“”にするのが目的なので，ERASEをFOR A＝0 TO1000：A(I)＝0：NEXTのようなものに置き換える。HuBASIC ではさらに，2度目のDIMを実行しないよう，DIMを取るかDIMを通らないようにプログラムを一部変更する。最後のERASEの目的は，要素の数をその場に応じて動的に割り当てることである。この場合は，考え得る最大の添字でDIMしておき，ERASEは取る。HuBASICで はさらにDIMを2回以上実行しないよう，プログラムの流れを変更する。	配列を消去する。	
	OPTION ANGLE DEGREESではSIN 等の関数とCIRCLE の角度の単位が「度」になる。	角度の単位を変更する。	
REM 〔注釈〕	S-BASICとSHARP BASICでは，注釈の中に：を入れることはできない。逆に言うと，それらでは，REM文の後に，：で区切って実行文を置くことができるが、その他ではできない。Hu→S等では：を取るか；などに変える。その逆ではマルチステートメントでなく，次の行に実行文を置く。	注釈を置く（ステートメントに続けるには前に：が必要)。	
	SHARP BASICでは，行の先頭ではREMに，行の途中では：REMに変える。	注釈を置く（ステートメントに続けるときも：は不要)。	
	この命令のないBASICでは単に取るだけで良い(まず問題はない)。	乱数の発生系列を変更する。	

97ページに続く

4.1 CLEAR CLR LIMIT SWAP MID$ PRINT LPRINT USING WRITE INPUT LINE INPUT LINPUT READ DATA RESTORE GOTO GOSUB RETURN IF～THEN～ELSE

分類	働き	N-BASIC (PC-8001ほか) Ⓓ…ディスク版にのみ存在	N88BASIC (PC-8801ほか) ⓈⓇ…88SRのサウンド関係の拡張命令 Ⓓ…ディスク版にのみ存在	MSX-BASIC V1.0 (各社MSX) Ⓓ…ディスク版にのみ存在
一般ステートメント	変数・配列・ユーザー関数等をクリアする。	CLEAR	CLEAR	CLEAR
	文字バッファの大きさを指定したりBASICの使用エリアを制限してマシン語エリアを確保したりする。	CLEAR 〔文字バッファの大きさ〕〔,アドレス〕	CLEAR 〔文字バッファの大きさ〕〔,〔アドレス〕〔,スタックの大きさ〕〕	CLEAR 〔文字バッファの大きさ〕〔,アドレス〕
	変数の内容を交換する。	SWAP 変数名，変数名	SWAP 変数名，変数名	SWAP 変数名，変数名
	文字変数の一部の文字列を置き換える。	MID$ (変数名,位置〔,長さ〕) ="文字列"	MID$ (変数名，位置〔,長さ〕)="文字列"	MID$ (変数名，位置〔,長さ〕)="文字列"
	画面にデータを表示する。	PRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]	PRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]	PRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]
	プリンタにデータを出力する。	LPRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]	LPRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]	LPRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]
		{PRINT / LPRINT} USING"書式";式[{; / ,}[式…]]	{PRINT / LPRINT} USING"書式";式 [{; / ,}[式…]]	{PRINT / LPRINT} USING "書式"; 式 [{; / ,} [式…]]
	画面にデータを表示する(PRINTと少し違う)。		WRITE [式[{; / ,}][式…]]] Ⓓ	
	キーボードよりデータを入力する（,で区切る)。	INPUT ["メッセージ";]変数名〔,変数名…〕	INPUT 〔WAIT 待ち時間,〕 ["メッセージ"{; / ,}] 変数名〔,変数名…〕	INPUT ["メッセージ";]変数名〔,変数名…〕
	キーボードよりデータを入力する（,で区切らない…文字変数専用)。	LINE INPUT ["メッセージ";]文字変数名	LINE INPUT 〔WAIT 待ち時間,〕 ["メッセージ"{; / ,}] 文字変数名	LINE INPUT ["メッセージ";]文字変数名
	プログラム中のデータを変数に読み込む。	READ 変数名〔,変数名…〕	READ 変数名〔,変数名…〕	READ 変数名〔,変数名…〕
	データを置く。	DATA〔データ〔,データ…〕〕	DATA〔データ〔,データ…〕〕	DATA〔データ〔,データ…〕〕
	データの読み取り位置を指定する。	RESTORE〔行番号〕	RESTORE〔行番号〕	RESTORE〔行番号〕
	指定行に実行を移す。	GOTO 行番号	GOTO 行番号	GOTO 行番号
	サブルーチンを呼び出す。	GOSUB 行番号	GOSUB 行番号	GOSUB 行番号
	サブルーチンから帰る。	RETURN	RETURN 〔行番号〕	RETURN〔行番号〕
	条件により処理を変える。	IF条件 {THEN {ステートメント / 行番号} / GOTO 行番号} [ELSE {ステートメント / 行番号}]	IF条件 {THEN {ステートメント / 行番号} / GOTO 行番号} [ELSE {ステートメント / 行番号}]	IF条件 [{THEN {ステートメント / 行番号} / GOTO 行番号}] [ELSE {ステートメント / 行番号}]

98ページに続く

F-BASIC V3.0 (FM-7ほか) Ⓓ…ディスク版にのみ存在	HuBASIC (MZ-700)	HuBASIC(X1・X1 turbo) ①CZ-8CB01 V1.0 ③CZ-8CB01 V2.0, CZ-8FB01 V2.0 ②CZ-8FB01 V1.0 ④CZ-8FB02 V1.0にのみ存在
CLEAR	CLEAR または CLR	CLEAR または CLR ①②④
CLEAR 〔文字バッファの大きさ〕〔,アドレス〕	CLEAR アドレス または LIMIT アドレス	CLEAR アドレスまたは LIMIT アドレス ①②④
SWAP 変数名,変数名	SWAP 変数名,変数名	SWAP 変数名, 変数名
MID$ (変数名,位置〔,長さ〕)="文字列"	MID$ (変数名,位置〔,長さ〕)="文字列"	MID$ (変数名, 位置〔, 長さ〕)="文字列"
PRINT [[{TAB(位置) SPC(スペース量) 式}][{; ,}]][…]	PRINT [[{TAB(位置) SPC(スペース量) 式}][{; ,}]][…]	PRINT [[{TAB(位置) SPC(スペース量) 式}][{; ,}]][…]
LPRINT [[{TAB(位置) SPC(スペース量) 式}][{; ,}]][…]	LPRINT [[{TAB(位置) SPC(スペース量) 式}][{; ,}]][…]	LPRINT [[{TAB(位置) SPC(スペース量) 式}][{; ,}]][…]
{PRINT LPRINT} USING"書式";式[[{; ,}][式…]]	{PRINT LPRINT} USING"書式"{; ,}[式[{; ,}][式…]]]	{PRINT LPRINT} USING"書式"{; ,}[式[{; ,}][式…]]]
	WRITE [式][{; ,}][式…]	WRITE〔式〕[{; ,}][式…]
INPUT ["メッセージ"{; ,}] 変数名〔,変数名…〕	INPUT ["メッセージ"{; ,}] 変数名〔,変数名…〕	INPUT ["メッセージ"{; ,}] 変数名〔,変数名…〕
LINE INPUT ["メッセージ"{; ,}] 文字変数名	{LINE INPUT LINPUT}["メッセージ"{; ,}] 文字変数名	{LINE INPUT LINPUT}["メッセージ"{; ,}] 文字変数名
READ 変数名〔,変数名…〕	READ 変数名〔,変数名…〕	READ 変数名〔,変数名…〕
DATA 〔データ〔,データ…〕〕	DATA 〔データ〔,データ…〕〕	DATA 〔データ〔, データ…〕〕
RESTORE 〔行番号〕	RESTORE 〔行番号〕	RESTORE 〔行番号〕
GOTO 行番号	GOTO 行番号	GOTO 行番号
GOSUB 行番号	GOSUB 行番号	GOSUB 行番号
RETURN 〔行番号〕	RETURN 〔行番号〕	RETURN 〔行番号〕
IF条件 {THEN {ステートメント 行番号} GOTO 行番号} [ELSE {ステートメント 行番号}]	IF条件 [{THEN {ステートメント 行番号} GOTO 行番号 GOSUB 行番号}] [ELSE {ステートメント 行番号}]	IF 条件 [{THEN {ステートメント 行番号} GOTO 行番号 GOSUB 行番号}][ELSE {ステートメント 行番号}]

99ページに続く

4.2 CLEAR CLR LIMIT SWAP MID$ PRINT LPRINT PRINT/P USING WRITE INPUT LINE INPUT READ DATA RESTORE GOTO GOSUB RETURN IF～THEN～ELSE

BASIC-M25 (MZ-2500)	BASIC-S25(MZ-2500)	S-BASIC(MZ-700/1500) ①MZ-700 テープ版(付属) ②MZ-700 QD・Disk版 ③MZ-1500QD・Disk版 にのみ存在
CLEAR	CLR	CLR
CLEAR {アドレス / MAX}	LIMIT {アドレス / MAX}	LIMIT {アドレス / MAX}
SWAP 変数名，変数名	SWAP 変数名，変数名	(代入文で代用)
MID$ (変数名，位置〔,長さ〕)="文字列"	MID$ (変数名，位置〔,長さ〕)="文字列"	
PRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]	PRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]	PRINT〔[〔文字色〕〔,背景色〕]〕 [{TAB (位置) / SPC(スペース量) / 式}][{; / ,}][…]
LPRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]	PRINT/P [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]	PRINT [{TAB(位置) / SPC(スペース量) / 式}][{; / ,}][…]
{PRINT / LPRINT} USING"書式"{; / ,}[式[{; / ,}[式…]]]	{PRINT / PRINT/P} USING"書式"{; / ,} [式[{; / ,}[式…]]]	PRINT {[〔文字色〕〔,背景色〕]〕 / PRINT/P} USING "書式"{; / ,}[式[{; / ,}[式…]]]
WRITE [式 [{; / ,}[式…]]]	WRITE [式 [{; / ,}[式…]]]	
INPUT 〔WAIT待ち時間,〕["メッセージ"{; / ,}] 変数名〔,変数名…〕	INPUT 〔WAIT 待ち時間,〕 ["メッセージ"{; / ,}] 変数名〔,変数名…〕	INPUT 〔"メッセージ; 〕変数名〔,変数名…〕
LINE INPUT 〔WAIT待ち時間,〕["メッセージ"{; / ,}]文字変数名	LINE INPUT 〔WAIT 待ち時間,〕 ["メッセージ"{; / ,}] 文字変数名	
READ 変数名 〔,変数名…〕	READ 変数名 〔,変数名…〕	READ 変数名 〔,変数名…〕
DATA 〔データ 〔,データ…〕〕	DATA 〔データ〔,データ…〕〕	DATA 〔データ〔,データ…〕〕
RESTORE 〔行番号〕	RESTORE 〔行番号〕	RESTORE 〔行番号〕
GOTO 行番号	GOTO 行番号	GOTO 行番号
GOSUB 行番号	GOSUB 行番号	GOSUB 行番号
RETURN 〔行番号〕	RETURN 〔行番号〕	RETURN
IF条件 [{THEN {ステートメント / 行番号} / GOTO 行番号 / GOSUB 行番号}] [ELSE {ステートメント / 行番号}]	IF条件 [THEN {ステートメント / 行番号} / GOTO 行番号 / GOSUB 行番号] [ELSE {ステートメント / 行番号}]	IF条件 {THEN {ステートメント / 行番号} / GOTO 行番号 / GOSUB 行番号} ① IF条件 [{THEN {ステートメント / 行番号} / GOTO 行番号 / GOSUB 行番号}]

100ページに続く

SHARP BASIC (MZ-80K/C/1200/80B/2000/2200) ⓪ SP-5030 ③ MZ-1Z001 ⑤ MZ-2Z001 ① SB-5520 ④ MZ-1Z002 ⑥ MZ-2Z021 ② SB-6520 にのみ存在	備考	働き	分類
CLR		変数・配列・ユーザー関数等をクリアする。	一般ステートメント
LIMIT {アドレス / MAX}	マイクロソフトBASICとM25では変数クリアを伴う。MAXとは可能な最大限のアドレス（&HFF00等）を示す。文字バッファとは，マイクロソフトBASICに特有なもので，文字変数の実内容が入る所を示す。N88以外では固定長なので最初に指定しておく。N88は可変だがその反面スタック（他では可変）は固定にしている（文字バッファの大きさの指定は互換性のためだけに用意されている）。国産系はすべて可変なのでこれらは取ってしまって良い。	文字バッファの大きさを指定したりBASICの使用エリアを制限してマシン語エリアを確保したりする。	
（代入文で代用）	SWAP A, BならC=A：A=B：B=Cのようにする（Cは使っていない変数を利用）。	変数の内容を交換する。	
	MID$（A$, B, C）=D$（Cが省略されていれば255と考える）なら，E=LEN(D$)：E=−(E>=C)＊C−(E<C)＊E： F=LEN(MID$（A$, B, 255)):E =−(E>=F)＊F−(E<F)＊E：A$=LEFT$(A$, B−1)+LEFT$(D$, E)+MID$(A$, B+E, 255)とする。条件によっては，これよりもっと簡単にすることもできるので，試してみると良い。	文字変数の一部の文字列を置き換える。	
PRINT [{TAB(位置) / 式} [{; / ,} TAB(位置) / 式 …]]] ※⓪ではSPCも使える（意味は他のものと同じ）	SHARP BASICの①～⑥SPCはSPACE$に置き換える。また⓪も含め，式と式等の間に；が入っていないときは，；を入れる（例PRINT SPC(10)123ABC→PRINT SPACE$(10)；123；ABC)。；は，S-BASICで後に関数が続くときも必要。マイクロソフトBASIC，Hu，M25では数値の後に自動的にスペースが1個入るがその他では入らないので注意（例 PRINT 123；-456は前者は␣123␣-456後者は␣123-456になる）。画面・プリンタのコントロールコード（CHR$（31以下）の結果）が各機種またはBASICまたはプリンタによって異なるので注意（特にS-BASIC・SHARP BASICとその他の間では全くと言っていいほど違う）。マイクロソフトBASICでは文字も数値も，M25では数値を画面に表示する際，次の行にまたがってしまう場合には，あらかじめ改行され，次の行の先頭から表示を始めるが、その他の場合はまたがって表示される。「，」を使ったとき，マイクロソフトBASICでは14桁ごと，その他では10桁ごとの位置から表示される。	画面にデータを表示する。	
PRINT/P [{TAB(位置) / 式} [{; / ,} [TAB(位置) / 式 …]]] ※⓪ではSPCも使える		プリンタにデータを出力する。	
{PRINT / PRINT/P} USING "書式" ；[式 [{; / ,} [式…]]]	SHARP BASICは倍精度BASIC（SB-5620，SB-6620，MZ-1Z003，MZ-2Z003）でのみ使用可能（書式はほぼ同じ）。Hu，M25，S25，S-BASICでは，PRINT，LPRINT（PRINT/P）の途中からUSING指定にすることもできる。		
	不要なスペースは詰め，「；」や「，」の所には「，」を置き（「；」と「，」は同意）文字の前後には，「"」を置く。WRITE 1，2；"3"なら1，2，"3"と表示される（PRINT文なら␣1␣…␣2␣3（…はスペース約10個）となる）。ただし，WRITEは画面に対してはあまり使われることがない。	画面にデータを表示する(PRINTと少し違う)。	
INPUT〔"メッセージ"；〕変数名〔，変数名…〕	SHARP BASICではメッセージを指定すると？␣は表示されない（メッセージは""でも良い）。メッセージを省略すると？␣は表示される。それ以外のBASICでは「"メッセージ"，」を指定した場合のみ？␣は表示されない。WAITの指定は主にデモ用。0.1秒単位で示す待ち時間が過ぎてもデータの入力がないときは次の行（マルチステートメントが続いても無視される）へ実行が移る。普通は取ってしまって良い。なお，実行時にただの↵(CR)を押すと，N-BASICやHuBASICでは値は保存され，次の文へ。N88とM25では変数が1個の場合のみ0または""が代入され，次の文へ。その他，データの個数や型が違う時の処理は各BASICによって違うので，実行時に正しい値を入力するよう，注意すること。	キーボードよりデータを入力する（，で区切る）。	
	S-BASIC，SHARP BASICではただのINPUT文にする。実行時，「，」を含むデータを入力するときは文字列全体を"～"で囲む。HuBASICではメッセージの分も変数に代入されてしまうので，LINE INPUT "DATA="，A$：A$=MID$（A$,6）のようにする。また，Huから他のBASICでは，この分の処理をカットする。	キーボードよりデータを入力する（，で区切らない…文字変数専用）。	
READ 変数名〔，変数名…〕		プログラム中のデータを変数に読み込む。	
DATA〔データ〔，データ…〕〕	SHARP BASIC以外では，データは16進定数等も置ける。SHARP BASICではDATA &HFFなどとあったら，DATA 255のように変えること。	データを置く。	
RESTORE〔行番号〕①～⑥		データの読み取り位置を指定する。	
GOTO 行番号		指定行に実行を移す。	
GOSUB 行番号		サブルーチンを呼び出す。	
RETURN	行番号を指定すると，その行に戻ることができる。つまりサブルーチンのスタックを一段下げてGOTOすることになる。S-BASICとSHARP BASICでは，XX=1：RETURNのようにし，呼び出したGOSUBの直後にIF XX=1 THEN 行番号とすること（GOSUBが何箇所あっても各々についてこれを行うこと）。	サブルーチンから帰る。	
IF条件 {THEN {ステートメント / 行番号} / GOTO 行番号 / GOSUB 行番号}	MZ-700のテープ版S-BASICとSHARP BASICでは，ELSE以降のあるIF文は変更の必要がある（ELSE以降は条件が成立しなかったときに実行される）。また，条件にANDやORが含まれていた場合はその両側の条件を（ ）でくくり，ANDは＊に，ORは＋にする。また，IF～THEN～ELSEがネストしているときは，ELSEは内側のIFに対応している。例えば， 100 IF A>B OR C=D THEN PRINT"A"ELSE IF E=F THEN PRINT "B" ELSE PRINT"C " は，	条件により処理を変える。	

101ページに続く

分類	働き	N-BASIC (PC-8001ほか) D…ディスク版にのみ存在	N88BASIC (PC-8801ほか) SR…88SRのサウンド関係の拡張命令 Ⓓ…ディスク版にのみ存在	MSX-BASIC V1.0 (各社MSX) Ⓓ…ディスク版にのみ存在
一般ステートメント				
	複行条件判断の最初の条件を置く。			
	複行条件判断の2番目〜最後の条件を置く。			
	複行条件判断のいずれの条件も合わなかった時の処理を置く。			
	複行条件判断の終了を示す。			
	式の値により実行を移す行を変える。	ON 数値式 GOTO 行番号〔,行番号…〕	ON 数値式 GOTO 行番号〔,行番号…〕	ON 数値式 GOTO 行番号〔,行番号…〕
	式の値により呼び出すサブルーチンを変える。	ON 数値式 GOSUB 行番号〔,行番号…〕	ON 数値式 GOSUB 行番号〔,行番号…〕	ON 数値式 GOSUB 行番号〔,行番号…〕
	式の値によりデータの読み取り位置を変える。			
	式の値によりサブルーチンから戻る行を変える。			
	指定回数のループを開始する。	FOR数値変数名＝初期値 TO 終了値〔STEP 増分〕	FOR 数値変数名＝初期値 TO 終了値〔STEP 増分〕	FOR 数値変数名＝初期値 TO 終了値〔STEP 増分〕
	FORループの終了位置を示す。	NEXT〔数値変数名〔,数値変数名…〕〕	NEXT〔数値変数名〔,数値変数名…〕〕	NEXT〔数値変数名〔,数値変数名…〕〕
	条件成立中ループを行う（最初から条件不成立なら一度もループしない）。		WHILE 条件式 Ⓓ	
	WHILEループの終了位置を示す。		WEND Ⓓ	
	条件成立までループを行う（最初から条件成立していても最低1度はループする）。			
	REPEATループの終了位置を示す。			
	プログラムの実行を中断する。	STOP	STOP	STOP
	プログラムの実行を終了する。	END	END	END
	プログラムの実行を一定時間停止する。			
	グラフィックRAMに配列を置く。			
	グラフィックRAM上の変数をクリアする。			
	ラベルを置く。		(単にラベル名だけで良い)	
	データをユーザースタックに積む。			
	データをユーザースタックから取る。			

102ページに続く

F-BASIC V3.0 (FM-7ほか) Ⓓ…ディスク版にのみ存在	HuBASIC (MZ-700)	HuBASIC (X1・X1 turbo) ①CZ-8CB01 V1.0 ③CZ-8CB01 V2.0, CZ-8FB01 V2.0 ②CZ-8FB01 V1.0 ④CZ-8FB02 V1.0にのみ存在
ON 数値式 GOTO 行番号〔,行番号…〕	ON 数値式 GOTO 行番号〔,行番号…〕	ON 数値式 GOTO 行番号〔,行番号…〕
ON 数値式 GOSUB 行番号〔,行番号…〕	ON 数値式 GOSUB 行番号〔,行番号…〕	ON 数値式 GOSUB 行番号〔,行番号…〕
	ON 数値式 RESTORE 行番号〔,行番号…〕	ON 数値式 RESTORE 行番号〔,行番号…〕
	ON 数値式 RETURN 行番号〔,行番号…〕	ON 数値式 RETURN 行番号〔,行番号…〕
FOR 数値変数名＝初期値 TO 終了値〔STEP 増分〕	FOR 数値変数名＝初期値 TO 終了値〔STEP 増分〕	FOR 数値変数名＝初期値 TO 終了値〔STEP増分〕
NEXT〔数値変数名〔,数値変数名…〕〕	NEXT〔数値変数名〔,数値変数名…〕〕	NEXT〔数値変数名〔,数値変数名…〕〕
WHILE 条件式	WHILE 条件式	WHILE 条件式
WEND	WEND	WEND
	REPEAT	REPEAT
	UNTIL 条件式	UNTIL 条件式
STOP	STOP	STOP
END	END	END
	PAUSE 時間	PAUSE 時間
		VDIM 配列名(添字の最大値〔,…〕)〔, 配列名…〕④
		VDIM CLEAR〔開始アドレス, 終了アドレス〕
	LABEL "ラベル名"	LABEL "ラベル名"
		PUSH 式〔, 式…〕
		POP 変数名〔, 変数名…〕

103ページに続く

5.2 IF～THEN ELSE IF ELSE END IF ON～GOTO ON～GOSUB ON～RESTORE ON～RETURN FOR NEXT WHILE WEND REPEAT UNTIL STOP END PAUSE LABEL

BASIC-M25（MZ-2500）	BASIC-S25（MZ-2500）	S-BASIC（MZ-700/1500） ①MZ-700 テープ版（付属） ②MZ-700 QD・Disk版 ③MZ-1500QD・Disk版 にのみ存在
		〔ELSE {ステートメント / 行番号}〕
IF 条件 THEN	IF 条件 THEN	
ELSE IF 条件 THEN	ELSE IF 条件 THEN	
ELSE	ELSE	
END IF	END IF	
ON 数値式 GOTO 行番号〔,行番号…〕	ON 数値式 GOTO 行番号〔,行番号…〕	ON 数値式 GOTO 行番号〔,行番号…〕
ON 数値式 GOSUB 行番号〔,行番号…〕	ON 数値式 GOSUB 行番号〔,行番号…〕	ON 数値式 GOSUB 行番号〔,行番号…〕
ON 数値式 RESTORE 行番号〔,行番号…〕	ON 数値式 RESTORE 行番号〔,行番号…〕	
ON 数値式 RETURN 行番号〔,行番号…〕	ON 数値式 RETURN 行番号〔,行番号…〕	
FOR 数値変数名＝初期値 TO 終了値〔STEP増分〕	FOR 数値変数名＝初期値 TO 終了値〔STEP増分〕	FOR 数値変数名＝初期値 TO 終了値〔STEP増分〕
NEXT〔数値変数名〔,数値変数名…〕〕	NEXT〔数値変数名〔,数値変数名…〕〕	NEXT〔数値変数名〔,数値変数名〕〕
WHILE 条件式	WHILE 条件式	
WEND	WEND	
REPEAT	REPEAT	
UNTIL 条件式	UNTIL 条件式	
STOP	STOP	STOP
END	END	END
PAUSE 時間	PAUSE 時間	
（単にラベル名だけで良い）	（単にラベル名だけで良い）	LABEL "ラベル名" ②③

104ページに続く

SHARP BASIC (MZ-80K/C/1200/80B/2000/2200) ⓪ SP-5030 ③ MZ-1Z001 ⑤ MZ-2Z001 ① SB-5520 ④ MZ-1Z002 ⑥ MZ-2Z021 ② SB-6520 にのみ存在	備　考	働　き	分類
	100 IF (A>B)+(C =D) THEN PRINT "A" : GOTO 110 102 IF E=F THEN PRINT "B" : GOTO 110 104 PRINT "C" 110 ～とする。		
	IF文の作用を複数の行に広げたもの。基本的な考え方は普通のIF～THEN～ELSEと変わらないので，GOTO文を使って簡単に置き換えることができる。なお，各THENの後とただのELSE（いずれの条件も不成立のとき）の後は，常に改行し，ELSEが行の先頭から始まるのが特徴。ELSE IF～THENとELSEはないこともあるがIF～THENとEND IFは必ず1対1で対応している。	複行条件判断の最初の条件を置く。	
		複行条件判断の2番目～最後の条件を置く。	
		複行条件判断のいずれの条件も合わなかった時の処理を置く。	
		複行条件判断の終了を示す。	
ON 数値式 GOTO 行番号〔,行番号…〕		式の値により実行を移す行を変える。	
ON 数値式 GOSUB 行番号〔,行番号…〕		式の値により呼び出すサブルーチンを変える。	
	これがないBASICではON～GOSUB で1度サブルーチンに飛び，その先でRESTORE して，RETURNするようにする。	式の値によりデータの読み取り位置を変える。	
	これがないBASICではただのRETURN 行番号のときのようにXX=1のようにフラグを立てていったんメインルーチンに戻り，そこでON～GOTOでジャンプするようにする。	式の値によりサブルーチンから戻る行を変える。	
FOR 数値変数名=初期値 TO 終了値〔STEP 増分〕	N88，Hu，M25では増分が正で初期値>終了値（FOR A=1 TO 0等）や増分が負で初期値<終了値の場合は1回もループしない。他のBASICでは少なくとも1度は実行してしまうので，FORループに入る前にIF文でテストし，場合によってはNEXTの次にジャンプする。逆にHu，M25への移植では問題にならないことが多いが，場合によってはNEXTの所で判断するIF 文に変えると良い。	指定回数のループを開始する。	一般ステートメント
NEXT〔数値変数名〔,数値変数名…〕〕	N88，Hu，M25 では上記の理由によりFOR文の所でループ脱出条件を判断するので，対応するNEXT はひとつしか認められない。つまりFORとNEXT は物理的に1対1に対応している必要がある。IF～THEN NEXT等は不可なので，THEN 行番号（NEXTのある行）のようにする。	FORループの終了位置を示す。	
	WHILEのないBASICでは，例えばWHILE A>Bなら，IF A=<B THEN 行番号（WENDの次の文のある行）のようにする。	条件成立中ループを行う（最初から条件不成立なら一度もループしない）。	
	WHILEのあった行にGOTOする。	WHILEループの終了位置を示す。	
	REPEATのないBASICでは単に取ってしまう。	条件成立までループを行う（最初から条件成立していても最低1度はループする）。	
	UNTILのないBASICでは，例えばUNTIL A=Bなら，IF A<>B THEN 行番号（REPEAT の次の文のある行）のようにする。	REPEATループの終了位置を示す。	
STOP		プログラムの実行を中断する。	
END	マイクロソフト，Hu，M25ではCLOSEの働きを含むので，他に移植する場合，OPENがあってCLOSEがないなら，ENDの直前にCLOSEを追加。	プログラムの実行を終了する。	
	時間はいずれも0.1秒単位の指定。PAUSEのないBASICではFOR～NEXTで代用（ループ回数は実験で決める）。	プログラムの実行を一定時間停止する。	
	X1turbo以外では一般のDIMに直す。逆に，X1turboではDIMでメモリ不足になってもVDIMでなら充分余るので，大量のデータを扱うプログラムの移植も容易になる。	グラフィックRAMに配列を置く。	
	VDIMで宣言した配列はCLEARやRUNでは消えないので，この文を使う。アドレスで範囲を指定してグラフィックRAMの一部のみを使用することができる（他のBASICに移植する場合は，V DIM CLEARは取ってしまって良い）。	グラフィックRAM上の変数をクリアする。	
		ラベルを置く。	
	実際に使われることは極めてまれだが，もし使われていたなら，仮想スタック用一次元配列を作る。もしスタックに置くデータの型が数値と文字の両方なら，文字配列にして数値を文字列で記憶させるか，マシン語エリアを確保し，スタックを構成させる。	データをユーザースタックに積む。	
		データをユーザースタックから取る。	

105ページに続く

分類	働き	N-BASIC（PC-8001ほか） Ⓓ…ディスク版にのみ存在	N88BASIC（PC-8801ほか） (SR)…88SRのサウンド関係の拡張命令 Ⓓ…ディスク版にのみ存在	MSX-BASIC V1.0（各社MSX） Ⓓ…ディスク版にのみ存在
一般ステートメント	エラー回復処理を行うことと，そのルーチンを宣言する。	ON ERROR GOTO 行番号	ON ERROR GOTO 行番号	ON ERROR GOTO 行番号
	エラー回復処理を行わないことを宣言する。	ON ERROR GOTO 0	ON ERROR GOTO 0	ON ERROR GOTO 0
	エラー処理ルーチンから復帰する。	RESUME [{行番号 | NEXT}]	RESUME [{行番号 | NEXT}]	RESUME [{行番号 | NEXT}]
	式の値によってエラー処理から復帰する行を変える。			
	強制的にエラーを発生させる。	ERROR エラー番号	ERROR エラー番号	ERROR エラー番号
ファイルステートメント	ファイルを書き込みモードで開く。	OPEN "ファイル名" FOR OUTPUT AS〔＃〕ファイル番号Ⓓ	OPEN "ファイル名" FOR OUTPUT AS〔＃〕ファイル番号Ⓓ	OPEN "ファイル名" FOR OUTPUT AS〔＃〕ファイル番号
	ファイルを読み出しモードで開く。	OPEN "ファイル名" FOR INPUT AS〔＃〕ファイル番号Ⓓ	OPEN "ファイル名" FOR INPUT AS〔＃〕ファイル番号Ⓓ	OPEN "ファイル名" FOR INPUT AS〔＃〕ファイル番号
	ファイルを追加書き込みモードで開く。	OPEN "ファイル名" FOR APPEND AS〔＃〕ファイル番号Ⓓ	OPEN "ファイル名" FOR APPEND AS〔＃〕ファイル番号Ⓓ	OPEN "ファイル名" FOR APPEND AS〔＃〕ファイル番号 Ⓓ
	ファイルをランダムアクセスモードで開く。	OPEN "ファイル名" AS〔＃〕ファイル番号Ⓓ	OPEN "ファイル名" AS〔＃〕ファイル番号Ⓓ	OPEN "ファイル名" AS〔＃〕ファイル番号Ⓓ
	ランダムファイルバッファに変数を割り当てる。	FIELD〔＃〕ファイル番号，長さ AS 文字変数名〔,長さ…〕Ⓓ	FIELD〔＃〕ファイル番号，長さ AS 文字変数名〔，長さ…〕Ⓓ	FIELD〔＃〕ファイル番号，長さ AS 文字変数名〔，長さ…〕Ⓓ
	ファイルを閉じる。	CLOSE〔〔＃〕ファイル番号〔，…〕〕Ⓓ	CLOSE〔〔＃〕ファイル番号〔，…〕〕	CLOSE〔〔＃〕ファイル番号〔，…〕〕
	ファイルにデータを出力する(「，」を特に付けない)。	PRINT ＃ ファイル番号 {[[{TAB(位置) | SPC(スペース量) | 式}]] [[{; | ,}]] [...] USING "書式"{; | ,}[式][{; | ,}][式…]]]}	PRINT＃ ファイル番号， {[[{TAB(位置) | SPC(スペース量) | 式}]][[{; | ,}]][...] USING "書式"{; | ,}[式][{; | ,}][式…]]]}	PRINT ＃ ファイル番号， {[[{TAB(位置) | SPC(スペース量) | 式}]][[{; | ,}]][...] USING "書式"{; | ,}[式[{; | ,}][式…]]]}
	ファイルにデータを出力する(データの区切りに「，」を付ける)。		WRITE〔＃〕ファイル番号，[式{[; | ,]}式…]]] Ⓓ	
	ファイルからデータを入力する(「，」をデータの区切りとみなす)。	INPUT ＃ ファイル番号，変数名〔，変数名…〕(Ⓓ)	INPUT ＃ ファイル番号，変数名〔，変数名…〕	INPUT ＃ ファイル番号， 変数名〔，変数名〕
	ファイルからデータを入力する(「，」をデータの区切りとみなさない)。	LINE INPUT ＃ファイル番号，文字変数名Ⓓ	LINE INPUT ＃ ファイル番号，文字変数名	LINE INPUT ＃ ファイル番号，文字変数名
	1行の長さを制限する(出力時)。		WIDTH {＃ファイル番号 | "デバイス名"}，長さ	

106ページに続く

F-BASIC V3.0 (FM-7ほか) D…ディスク版にのみ存在	HuBASIC (MZ-700)	HuBASIC (X1・X1 turbo) ①CZ-8CB01 V1.0 ③CZ-8CB01 V2.0, CZ-8FB01 V2.0 ②CZ-8FB01 V1.0 ④CZ-8FB02 V1.0にのみ存在
ON ERROR GOTO 行番号	ON ERROR GOTO 行番号	ON ERROR GOTO 行番号
ON ERROR GOTO 0	ON ERROR GOTO 0	ON ERROR GOTO 0
RESUME [{行番号 / NEXT}]	RESUME [{行番号 / NEXT}]	RESUME [{行番号 / NEXT}]
	ON 数値式 RESUME 行番号〔, 行番号…〕	ON 数値式 RESUME 行番号〔, 行番号…〕
ERROR エラー番号	ERROR エラー番号	ERROR エラー番号
OPEN "O",〔#〕ファイル番号,"ファイル名"	OPEN "O",〔#〕ファイル番号,"ファイル名"	OPEN "O",〔#〕ファイル番号,"ファイル名"
OPEN "I",〔#〕ファイル番号,"ファイル名"	OPEN "I",〔#〕ファイル番号,"ファイル名"	OPEN "I",〔#〕ファイル番号,"ファイル名"
OPEN "A",〔#〕ファイル番号,"ファイル名"		OPEN "A",〔#〕ファイル番号,"ファイル名"②③④
OPEN "R",〔#〕ファイル番号,"ファイル名"		OPEN "R",〔#〕ファイル番号,"ファイル名"②③④
FIELD〔#〕ファイル番号, 長さ AS 文字変数名〔, 長さ…〕Ⓓ		FIELD〔#〕ファイル番号, 長さ AS 文字変数名〔, 長さ…〕②③④
CLOSE〔〔#〕ファイル番号〔, …〕〕	CLOSE〔〔#〕ファイル番号〔, …〕〕	CLOSE〔〔#〕ファイル番号〔, …〕〕
PRINT # ファイル番号 { [{TAB(位置) / SPC(スペース量) / 式}] [{; / ,}] […] USING "書式" {; / ,} [式 [{; / ,} [式…]]] }	PRINT # ファイル番号, [{ [{TAB(位置) / SPC(スペース量) / 式}] [{; / ,}] […] USING "書式" {; / ,} [式 [{; / ,} [式…]]] }]	PRINT # ファイル番号 [{ [{TAB(位置) / SPC(スペース量) / 式}] [{; / ,}] […] USING "書式" {; / ,} [式 [{; / ,} [式…]]] }]
	WRITE # ファイル番号[, 式 [{; / ,} [式…]]]	WRITE # ファイル番号[, 式 [{; / ,} [式…]]]
INPUT # ファイル番号, 変数名,〔変数名〕	INPUT # ファイル番号, 変数名〔, 変数名…〕	INPUT # ファイル番号, 変数名〔, 変数名…〕
LINE INPUT # ファイル番号, 文字変数名	{LINE INPUT / LINPUT} #ファイル番号, 文字変数名	{LINE INPUT / LINPUT} #ファイル番号, 文字変数名

107ページに続く

6.2 ON ERROR GOTO RESUME ON～RESUME ERROR OPEN WOPEN ROPEN AOPEN XOPEN FIELD CLOSE PRINT# PRINT/T WRITE# INPUT# INPUT /T LINEINPUT#

BASIC-M25 (MZ-2500)	BASIC-S25(MZ-2500)	S-BASIC(MZ-700/1500) ①MZ-700 テープ版(付属) ②MZ-700 QD・Disk版 ③MZ-1500QD・Disk版 にのみ存在
ON ERROR GOTO 行番号	ON ERROR GOTO 行番号	ON ERROR GOTO 行番号
ON ERROR GOTO 0	ON ERROR GOTO 0	ON ERROR GOTO 0
RESUME [{行番号 / NEXT}]	RESUME [{行番号 / NEXT}]	RESUME [{行番号 / NEXT}]
ON 数値式 RESUME 行番号〔, 行番号…〕	ON 数値式 RESUME 行番号〔, 行番号…〕	
ERROR エラー番号	ERROR エラー番号	ERROR エラー番号
OPEN "O", # ファイル番号, "ファイル名"	WOPEN # ファイル番号, "ファイル名"	WOPEN "ファイル名" ① WOPEN # ファイル番号, "ファイル名"②③
OPEN "I", # ファイル番号, "ファイル名"	ROPEN # ファイル番号, "ファイル名"	ROPEN("ファイル名") ① ROPEN # ファイル番号, "ファイル名"②③
OPEN "A", # ファイル番号, "ファイル名"	AOPEN # ファイル番号, "ファイル名"	
OPEN "R", # ファイル番号, "ファイル名"〔, レコード長〕	XOPEN # ファイル番号, "ファイル名"〔, レコード長〕	XOPEN # ファイル番号, "ファイル名"②③
FIELD # ファイル番号, 長さ AS 文字変数名〔, 長さ…〕	FIELD # ファイル番号, 長さ AS 文字変数名〔, 長さ…〕	
CLOSE〔#ファイル番号〔, …〕〕	CLOSE〔#ファイル番号〔, …〕〕	CLOSE ① CLOSE〔#ファイル番号〔, …〕〕②③
PRINT # ファイル番号 { [[{TAB(位置) / SPC(スペース量) / 式}][{; / ,}]][…] / , USING "書式" {; / ,}[式[{; / ,}[式…]]] }	PRINT # ファイル番号 { [, 式{; / ,}[…]] / , USING "書式" {; / ,}[式[{; / ,}[式…]]]{; / ,}[式[{; / ,}[式…]]] }	PRINT { /T① / #ファイル番号, ②③ } { [, 式[{; / ,}][…]] / USING "書式"{; / ,}[式[{; / ,}[式…]]] }
WRITE # ファイル番号 [, 式[{; / ,}[式…]]]		
INPUT # ファイル番号, 変数名〔, 変数名…〕	INPUT # ファイル番号, 変数名〔, 変数名…〕	INPUT/T変数名〔, 変数名…〕①
LINE INPUT # ファイル番号, 文字変数名	LINE INPUT # ファイル番号, 文字変数名	INPUT # ファイル番号, 変数名〔, 変数名…〕

108ページに続く

SHARP BASIC (MZ-80K/C/1200/80B/2000/2200) ⓪ SP-5030　③ MZ-1Z001　⑤ MZ-2Z001 ① SB-5520　④ MZ-1Z002　⑥ MZ-2Z021 ② SB-6520　にのみ存在	備　　考	働　き	分類
ON ERROR GOTO 行番号②⑤⑥	テープ版のSHARP BASICでは根本的な代用法はないが，INPUT等では極力適正な値を入力するように気を付けたり，エラーになる前にデータ等をチェックするようにして，問題があればエラー回復ルーチンをGOSUBで呼び出す。また，どうしてもというなら，MZ-2Z001（⑤）と同内容のBASICがMZ-2000・2200用に出ているので，これを使っても良い。型番はMZ-1Z010。	エラー回復処理を行うことと，そのルーチンを宣言する。	一般ステートメント
ON ERROR GOTO 0②⑤⑥	エラー回復中では，そこで回復処理を中断し，コマンドレベルに戻る。	エラー回復処理を行わないことを宣言する。	一般ステートメント
RESUME [{行番号 / NEXT}]②⑤⑥	ただのRESUMEではエラー発生文に，NEXTを付けるとその次の文に、行番号を付けるとその行にジャンプさせる。	エラー処理ルーチンから復帰する。	一般ステートメント
	RESUME 行番号の変形。ON～GOTO　でジャンプした先でRESUME　行番号を利用する。	式の値によってエラー処理から復帰する行を変える。	一般ステートメント
	エラー番号は各BASICによって多少異なるが，この命令はエラー回復ルーチンのデバッグや未定義のエラー番号を利用したユーザーエラーを起こすもの。前者はプログラム中には出てこない。後者はSHARP BASICではGOSUBにできないか試す。	強制的にエラーを発生させる。	一般ステートメント
WOPEN〔 /T 〕"ファイル名"①③④(SP-5030は/Tは付けない) WOPEN {/T / #ファイル番号,} "ファイル名" ②⑤⑥	マイクロソフト，Huでは既にそのファイルが登録されているときはそれを消して新たに作成する(M25, S25, S-BASIC, SHARP BASIC では既に存在するならKILL・DELETEする)。ファイル番号はN, N88, MSX, HuではHow many files ? かMAX FILEで指定した番号以下で指定する。他ではファイル番号は1～100程度なら自由に決めて良く，同時に開く個数のみ制限される（論理ファイル番号→物理ファイル番号へのテーブルを持っている)。	ファイルを書き込みモードで開く。	ファイルステートメント
ROPEN〔/T〕〔"ファイル名"〕①③④ (SP-5030は/Tは付けない) ROPEN {/T〔"ファイル名"〕 / #ファイル番号, "ファイル名} ②⑤⑥	N, N88, MSX, Huでは同じファイルをいくつかのファイル番号で同時に開くことができる。	ファイルを読み出しモードで開く。	ファイルステートメント
	これがないBASICでは，ファイルを読み出し，別のファイルに書き込み，読み出した方のみCLOSEする。そうすれば，ファイルの末尾からデータを追加してゆくことが可能である。	ファイルを追加書き込みモードで開く。	ファイルステートメント
XOPEN # ファイル番号, "ファイル名"②⑤⑥	ランダムアクセスの手法はシャープ系とマイクロソフト系では大きく異なる。	ファイルをランダムアクセスモードで開く。	ファイルステートメント
	マイクロソフト系BASICではデータは256(M25はもっと小さくできる）文字分のバッファを介してファイルとデータの交換を行う。つまりGETによりバッファに入った1レコード分のデータは，このFIELDで割り当てた文字変数に自動的に代入される。その変数は一般の文字変数と全く同じように参照して良い。逆にFIELDで割り当てた変数にデータを代入する(LSET・RSETを使う)と，自動的にバッファにそのデータがセットされる。そしてPUTを行うとファイルにデータが書き込まれる。FIELD #1,200 ASA$, 40ASB$, 16ASC$なら1レコードは200, 40, 16文字の3分割して使われる。	ランダムファイルバッファに変数を割り当てる。	ファイルステートメント
CLOSE〔/T〕 ①③④ CLOSE [{/T / #ファイル番号〔, …〕}] ②⑤⑥		ファイルを閉じる。	ファイルステートメント
PRINT {/T / #ファイル番号, ②⑤⑥} {[式 / ,] [{; / ,}] […] / USING "書式" {; / ,} [式… [{; / ,} [式…]]]}	マイクロソフト，Hu，M25 では画面への出力と全く同じイメージで書き込まれるが,S-BASICのQD, ディスク版・S25では「，」が使われるとそこに改行コード(0DH)が書き込まれる。 S-BASICのテープ版とSHARP BASICでは「；」も「，」も改行コードになる。改行コードになるものをそうならないものに移植する時は，念のため2つ以上のPRINT文にしたほうが良い（PRINT #1, A$, B$→PRINT #1, A$: PRINT # 1, B$ のように)。逆にHu等の文字の「；」は文字列を結合して出力する(PRINT #1, A$; B$→PRINT #1, A$＋B$等)。	ファイルにデータを出力する(「，」を特に付けない)。	ファイルステートメント
	INPUT # 文と組み合わせると，相性も良く，容量の効率も優れている。ただ，これがないBASICではPRINT # 文にして良い(「；」は「，」にすること)。WRITE文の所も参照すること（同じイメージである)。	ファイルにデータを出力する(データの区切りに「，」を付ける)。	ファイルステートメント
	マイクロソフト，Huでは，数値の代入時，「，」に加えスペースもデータの区切りとみなすので，PRINT時「；」で区切られた数値データを，分けて入力することができる。他ではそれができないので，QD・ディスク版のS-BASICとS25ではPRINT文の「；」があれば，「，」にする。M25では，ひとつのPRINT文ではひとつデータしか出力しないようにする(「 ；」や「，」で分けて2つ以上のPRINTにする)。Nでは#－1としてカセットをアクセス可(PRINT # でも同様)。OPENは不用なので移植時にOPENを追加すること。	ファイルからデータを入力する(「，」をデータの区切りとみなす)。	ファイルステートメント
INPUT {/T / #ファイル番号, ②⑤⑥} 変数名〔, 変数名…〕	PRINT # 文と組み合わせると，文字データの受け渡しに向いている(CHR$(32)未満がない限り，そのままそっくり保存しておける)。	ファイルからデータを入力する(「，」をデータの区切りとみなさない)	ファイルステートメント
	長さを超える前に自動的に改行コードが入る。	1行の長さを制限する(出力時)。	ファイルステートメント

109ページに続く

分類	働き	N-BASIC（PC-8001ほか） Ⓓ…ディスク版にのみ存在	N88BASIC（PC-8801ほか） (SR)…88SRのサウンド関係の拡張命令 Ⓓ…ディスク版にのみ存在	MSX-BASIC V1.0（各社MSX） D…ディスク版にのみ存在
ファイルステートメント	ランダムファイルバッファに左づめでデータを置く。	LSET 文字変数名＝文字式 Ⓓ	LSET 文字変数名＝"文字式" Ⓓ	LSET 文字変数名＝"文字式" Ⓓ
	ランダムファイルバッファに右づめでデータを置く。	RSET 文字変数名＝文字式 Ⓓ	RSET 文字変数名＝"文字式" Ⓓ	RSET 文字変数名＝"文字式" Ⓓ
	ランダムファイルに1レコード出力する。	PUT〔#〕ファイル番号〔,レコード番号〕Ⓓ	PUT〔#〕ファイル番号〔,レコード番号〕Ⓓ	PUT〔#〕ファイル番号〔,レコード番号〕Ⓓ
	ランダムファイルから1レコード入力する。	GET〔#〕ファイル番号〔,レコード番号〕Ⓓ	GET〔#〕ファイル番号〔,レコード番号〕Ⓓ	GET〔#〕ファイル番号〔,レコード番号〕Ⓓ
	同時に開けるファイルの数を指定する。	（起動時にHow many files？で答える）Ⓓ	（起動時にHow many files？で答える）Ⓓ	MAXFILES＝ファイル数
	ダイレクトにディスクから入力する。	DSKI$（ドライブ番号,シリンダ番号,面番号,セクタ番号）(関) Ⓓ	DSKI$（ドライブ番号,シリンダ番号,面番号,セクタ番号）(関) Ⓓ	DSKI$（ドライブ番号,シリンダ番号,面番号,セクタ番号）(関) Ⓓ
	ダイレクトにディスクを出力する。	DSKO$ ドライブ番号,シリンダ番号,面番号,セクタ番号 Ⓓ	DSKO$ ドライブ番号,シリンダ番号,面番号,セクタ番号 Ⓓ	DSKO$ ドライブ番号,シリンダ番号,面番号,セクタ番号 Ⓓ
	デバイスを初期化する。	FORMAT ドライブ番号 Ⓓ		CALL FORMAT Ⓓ
	ファイルを消去する。	KILL "ファイル名" Ⓓ	KILL "ファイル名" Ⓓ	KILL "ファイル名" Ⓓ
	ファイル名を付け換える。	NAME "旧ファイル名" AS "新ファイル名" Ⓓ	NAME "旧ファイル名" AS "新ファイル名" Ⓓ	NAME "旧ファイル名" AS "新ファイル名" Ⓓ
	ファイルを書き込み禁止にする。	SET ｛ドライブ番号 "ファイル名" #ファイル番号｝,"P" Ⓓ	SET ｛ドライブ番号 "ファイル名" #ファイル番号｝,"P" Ⓓ	
	ファイルの書き込み時ベリファイするようにする。	SET ｛ドライブ番号 "ファイル名" #ファイル番号｝,"R" Ⓓ	SET ｛ドライブ番号 "ファイル名" #ファイル番号｝,"R" Ⓓ	
	ファイルを通常の状態に戻す。	SET ｛ドライブ番号 "ファイル名" #ファイル番号｝," " Ⓓ	SET ｛ドライブ番号 "ファイル名" #ファイル番号｝,"" Ⓓ	
	ファイルを不可視状態にする（ディレクトリを取っても見えないようにする）。			
	デバイスを選択する。			
	階層ディレクトリを選択する。			
	階層ディレクトリを作成する。			
	階層ディレクトリを消去する。			
	ファイルをコピーする。			COPY "転送元ファイル名" TO "転送先ファイル名" Ⓓ
	ハードディスクのヘッドを解放する。			
	ファイルを閉じずにファイルバッファを解放する。			
	ドライブの種類を選択する。			

110ページに続く

F-BASIC V3.0（FM-7ほか） Ⓓ…ディスク版にのみ存在	HuBASIC（MZ-700）	HuBASIC(X1・X1 turbo) ①CZ-8CB01 V1.0 ③CZ-8CB01 V2.0，CZ-8FB01 V2.0 ②CZ-8FB01 V1.0 ④CZ-8FB02 V1.0にのみ存在
LSET 文字変数名＝“文字式”Ⓓ		LSET 文字変数名＝“文字式” ②③④
RSET 文字変数名＝“文字式”Ⓓ		RSET 文字変数名＝“文字式” ②③④
PUT〔＃〕ファイル番号〔,レコード番号〕Ⓓ		PUT〔＃〕ファイル番号〔，レコード番号〕②③④
GET〔＃〕ファイル番号〔,レコード番号〕Ⓓ		GET〔＃〕ファイル番号〔，レコード番号〕②③④
（起動時に How many files ?で答える）Ⓓ	MAXFILES ファイル数	MAXFILES ファイル数
DSKI$(ドライブ番号,トラック番号， セクタ番号）㊐Ⓓ		DEVI $ “デバイス名”,レコード番号，文字変数名,文字変数名
DSKO $ ドライブ番号,トラック番号,セクタ番号Ⓓ		DEVO $ “デバイス名”,レコード番号,“文字データ”,“文字データ”
DSKINI ドライブ番号 Ⓓ		INIT “デバイス名”
KILL “ファイル名” Ⓓ		KILL “ファイル名”
NAME “旧ファイル名” AS “新ファイル名” Ⓓ		NAME “旧ファイル名”AS “新ファイル名” ②③④
		SET {“ファイル名” / ＃ファイル番号}，“P” ②③④
		SET {“ファイル名” / ＃ファイル番号}，“R” ②③④
		SET {“ファイル名” / ＃ファイル番号}，“ ” ②③④
		SET “ファイル名”，“S” ④
	DEVICE “デバイス名”	DEVICE “デバイス名”
		CHDIR “ディレクトリ名” ④
		MKDIR “ディレクトリ名” ④
		RMDIR “ディレクトリ名” ④
		COPY “転送元ファイル名” TO “転送先ファイル名” ④
		HDOFF ドライブ番号④

111ページに続く

7.2 LSET RSET PUT# PRINT# GET# INPUT# DEVI$ DEVO$ INIT KILL DELETE NAME RENAME SET LOCK UNLOCK VERIFY CHDIR DEFAULT MKDIR RMDIR

BASIC-M25 (MZ-2500)	BASIC-S25(MZ-2500)	S-BASIC(MZ-700/1500) ①MZ-700 テープ版(付属) ②MZ-700 QD・Disk版 ③MZ-1500QD・Disk版 にのみ存在
LSET 文字変数名＝"文字式"	LSET 文字変数名＝"文字式"	
RSET 文字変数名＝"文字式"	RSET 文字変数名＝"文字式"	
PUT #ファイル番号〔,レコード番号〕	{PUT # ファイル番号〔,レコード番号〕 PRINT # ファイル番号(レコード番号),式〔,式…〕}	PRINT # ファイル番号(レコード番号), 式〔,式…〕②③
GET #ファイル番号〔,レコード番号〕	{GET # ファイル番号〔,レコード番号〕 INPUT # ファイル番号(レコード番号),変数名〔,変数名…〕}	INPUT # ファイル番号(レコード番号),式〔,式…〕②③
(常に17個まで開けるので不要)	(常に17個まで開けるので不要)	(必要に応じて動的に割り当てられるので不要)②③
DEVI$ "デバイス名",レコード番号,文字変数名,文字変数名	DEVI$ "デバイス名",レコード番号, 文字変数名, 文字変数名	
DEVO$ "デバイス名",レコード番号,"文字データ","文字データ"	DEVO$ "デバイス名",レコード番号,"文字データ","文字データ"	
INIT "デバイス名"	INIT "デバイス名"	INIT "デバイス名" ②③
KILL "ファイル名"	DELETE "ファイル名"	DELETE "ファイル名" ②③
NAME "旧ファイル名" AS "新ファイル名"	RENAME "旧ファイル名","新ファイル名"	RENAME "旧ファイル名","新ファイル名"
SET "ファイル名","P"	LOCK "ファイル名"	LOCK "ファイル名" ②③
VERIFY {ON / OFF}	VERIFY {ON / OFF}	
SET "ファイル名",""	UNLOCK "ファイル名"	UNLOCK "ファイル名" ②③
CHDIR "デバイス名"	CHDIR "デバイス名"	DEFAULT "デバイス名" ②③
CHDIR "ディレクトリ名"	CHDIR "ディレクトリ名"	
MKDIR "ディレクトリ名"	MKDIR "ディレクトリ名"	
RMDIR "ディレクトリ名"	RMDIR "ディレクトリ名"	
	KILL 〔#ファイル番号〔,…〕〕	KILL 〔#ファイル番号〔,…〕〕
INIT"〔FD〕番号：{ドライブタイプ / シリンダ数,ステップレート,タイミング}"	INIT "〔FD〕番号：{ドライブタイプ / シリンダ数,ステップレート,タイミング}"	

112ページに続く

SHARP BASIC (MZ-80K/C/1200/80B/2000/2200) ⓪ SP-5030 ③ MZ-1Z001 ⑤ MZ-2Z001 ① SB-5520 ④ MZ-1Z002 ⑥ MZ-2Z021 ② SB-6520 にのみ存在	備　考	働　き	分類
	文字式の長さがFIELDで指定した長さより大きいときは，左のほうから順に採用され，右のほうのデータは切り捨てられる。文字列の長さが不足のときは左づめで代入され，余った所（右のほう）にはスペースが入る。	ランダムファイルバッファに左づめでデータを置く。	ファイルステートメント
	文字式の長さがFIELDで指定した長さより大きいときは，左のほうから順に採用され,右のほうのデータは切り捨てられる。文字列の長さが不足のときは右づめで代入され,余った所(左のほう）にはスペースが入る。一般にはLSETがよく使われ，RSETはあまり使われない。	ランダムファイルバッファに右づめでデータを置く。	
PRINT ＃ ファイル番号(レコード番号), 式〔,式〕②⑤⑥	シャープ系のBASICでは1レコードは32文字分（S 25ではもっと増やせる）の固定で，1レコードに1つのデータを置くようにしている。ファイルバッファの概念がないため，取っ付きやすいものの，文字列が33文字以上の時に面倒なのと，数値データの効率が良くない（マイクロソフト系ではMKS$，CVS 等の関数を使えば内部表現のまま数値を置ける)。このためシャープ系←→ マイクロソフト系では,プログラムと共に,データの構造をも大幅に変更する必要がある。	ランダムファイルに1レコード出力する。	
INPUT ＃ ファイル番号(レコード番号)， 式〔,式〕②⑤⑥		ランダムファイルから1レコード入力する。	
（必要に応じて動的に割り当てられるので不要）②⑤⑥		同時に開けるファイルの数を指定する。	
	マイクロソフトでは，データの受け渡しにはファイルバッファの＃0（LOAD・SAVE 等用でデータファイル用には使われない）が使われ，FIELD ＃0～でランダムファイル同様の方法で割り当てられた変数を介して入出力する。なおマイクロソフトではディスク専用だがHu等では他のデバイスが使える。 N，MSXは物理フォーマット。F，Huは論理フォーマット。その他も含め，これらの間の互換性は特にない。	ダイレクトにディスクから入力する。	
		ダイレクトにディスクを出力する。	
		デバイスを初期化する。	
DELETE "ファイル名" ②⑤⑥		ファイルを消去する。	
RENAME "旧ファイル名", "新ファイル名" ②⑤⑥		ファイル名を付け換える。	
LOCK "ファイル名" ②⑤⑥	N,N_{88}はドライブ全体に対しても行える。主に信頼性向上のためのものなので，当てはまる機能のないBASICでは，取ってしまって良い。	ファイルを書き込み禁止にする。	
		ファイルの書き込み時ベリファイするようにする。	
UNLOCK "ファイル名" ②⑤⑥		ファイルを通常の状態に戻す。	
	ないものでは取ってしまって良い。	ファイルを不可視状態にする（ディレクトリを取っても見えないようにする)。	
	各ステートメントでデバイス名を省略したときのデバイスを選択するもの。例えばHuでDEVICE "1:" とすると，次にOPEN "I",＃1,"TEST" としたときにはOPEN"I", ＃1, "1:TEST" を意味するようになる。	デバイスを選択する。	
	X1 turbo, MZ-2500以外の機種では，とりあえず取ってしまう(あまり使われることはないが)。	階層ディレクトリを選択する。	
		階層ディレクトリを作成する。	
		階層ディレクトリを消去する。	
	プログラムなら一度ロードしてセーブするようにする。データなら簡単なユーティリティを作って代用する。	ファイルをコピーする。	
	X1turbo以外はハードディスクが使えないので関係ない。	ハードディスクのヘッドを解放する。	
KILL〔＃ファイル番号〔, …〕〕	一度CLOSEした後，そのファイルが書き込みモードだったときは，KILL文でファイルを削除する(この文は,何らかの理由でファイルとして正式に登録する必要のない場合のキャンセル用に使われる)。	ファイルを閉じずにファイルバッファを解放する。	
	旧タイプ，新タイプのディスクドライブに対応するための命令(他のパソコンでは取って良い)。	ドライブの種類を選択する。	

113ページに続く

分類	働き	N-BASIC (PC-8001ほか) Ⓓ…ディスク版にのみ存在	N88BASIC (PC-8801ほか) ⓈⓇ…88SRのサウンド関係の拡張命令 Ⓓ…ディスク版にのみ存在	MSX-BASIC V1.0 (各社MSX) D…ディスク版にのみ存在
ファイル操作ステートメント	RAMディスクを初期化する。			
	RS-232Cのパラメータを設定する。	INIT %番号, モード, コマンド ※モード：i8251のコマンドポートに書き込む値	OPEN"COM番号:〔パリティ ビット〔ストップビット〔制御方法〔カナ表現〕〕〕〕"AS〔#〕ファイル番号 ※パリティ:E(偶数) N(なし) 0(奇数) のどれか1文字。 ビット:7または8の数字 (8の時はパリティにEか0は指定できない)。 ストップビット:1(1ビット) 2(1.5ビット) 3(2ビット)のどれか1文字。 制御方法:X(XON/XOFF制御する) N(しない) のどちらか。 カナ表現:S(ビットが7のときに有効で, SI/SOによりカナを送受信する) N(しない) のどちらか。	
	RS-232Cにデータを送信する。	PRINT %番号, 式〔…〕	(PRINT # 等を使用)	
	RS-232Cからデータを受信する。	INPUT %番号, 変数〔, …〕	(INPUT # 等を使用)	
画面モード・文字画面制御	文字画面のモードを設定する(主に表示桁数)。	WIDTH 〔桁数〕〔, 行数〕 ※桁数は36, 40, 72, 80のいずれか。行数は20または25。擬似グラフィックのドット数はこの横2倍, 縦4倍に連動して変わる。WIDTH 80, 25なら160×100ドットになる。	WIDTH 桁数〔, 行数〕 ※桁数は40または80。行数は20または25。グラフィック画面は変わらない(非連動)。	(SCREENで設定)。
	文字画面のモードを設定する (主にスクロール範囲)。	CONSOLE 〔開始行〕〔, 〔行数〕〔, 〔ファンクションキー表示〕〔, カラーモード〕〕〕 ※カラーモードが0ならモノクロモードになり, COLOR文の指定はブリンク等になる。1ならカラーモードになり, COLOR文の指定は色指定になる。ファンクションキー表示は0なら表示しない, 1なら画面の最下行に表示する (実際に文字が表示できる部分は減る)。	CONSOLE〔開始行〕〔〔行数〕〔, 〔ファンクションキー表示〕〔, カラーモード〕〕〕 ※Nと同じ。	WIDTH 桁数 ※32桁モードでWIDTH 20を実行すると, 7桁～26桁目が範囲になる。つまり, 左右の端から表示範囲がせばめられてゆく。

F-BASIC V3.0（FM-7ほか） Ⓓ…ディスク版にのみ存在	HuBASIC（MZ-700）	HuBASIC（X1・X1 turbo） ①CZ-8CB01 V1.0　③CZ-8CB01 V2.0，CZ-8FB01 V2.0 ②CZ-8FB01 V1.0　④CZ-8FB02 V1.0にのみ存在
		INIT "MEM〔番号〕:"
OPEN "{ I ／ 0 }",〔#〕ファイル番号, "COM番号：〔(クロック ビット パリティ ストップビット)〕" ※クロック：S($\frac{1}{64}$) またはF($\frac{1}{16}$)のどちらか。 2種類のボーレイトがビット：7または8の数字。 パリティ：E(偶数)N(なし)O(奇数)のどれか1文字 ストップビット：1(1ビット) 2(1.5ビット) 3(2ビット)のどれか1文字		OPEN"C",〔#〕ファイル番号, "COM番号：ボーレイトコード パリティ ビット ストップビット〔制御方法〔カナ表現〔送信改行指定〔受信改行指定〔日本語表現〔エンドコード〕〕〕〕〕〕" ④ ※ボーレイトコード：0～6(0=150, 1=300, 2=600, 3=1200, 4=2400, 5=4800, 6=9600)の数字 パリティ：E(偶数)N(なし)O(奇数)のいずれか ビット：5, 6, 7, 8, のいずれか ストップビット：1(1ビット)2(1.5ビット)3(2ビット) のいずれか 制御方法：X(XON/XOFF制御する)R(RTSでXON/XOFFする) Nまたは省略(しない)のいずれか カナ表現：S(SI/SO制御する)Nまたは省略(しない)のいずれか 改行指定：C(CRのみ)Lまたは省略(CR+LF)のいずれか 日本語表現：J(PC-9801コンパチの$1B_H+4B_H/1B_H+48_H$で制御) Nまたは省略(シフトJIS)のいずれか エンドコード：キャラクタコード00_H～$1F_H$のいずれかに40_Hか60_Hを加えた文字。Dまたはdなら04_Hが終了コードになる
(PRINT # 等を使用)		(PRINT # 等を使用)
(INPUT # 等を使用)		(INPUT # 等を使用)
WIDTH〔桁数〕〔, 行数〕 ※桁数は40または80。行数は20または25。グラフィック非連動。		WIDTH 桁数 ①②③ WIDTH〔桁数〕〔, 行数〕〔, グラフィック縦解像度〕〔, ディスプレイモード〕〕〕④ ※桁数は40か80。行数は10, 12, 20, 25のいずれか。グラフィック縦解像度は0なら192または200ドット, 1なら384または400ドットになる(X方向は桁数×8ドット)。ディスプレイモードは0または省略で前面パネルのスイッチの状態に従う, 1で15.75kHzタイプディスプレイ, 2で24.5kHzタイプ(高解像度)ディスプレイ。ただしグラフィック縦400(384)ドットは高解像度のみ可。また行数10行は低解像ディスプレイのみ可。行数が20, 10行時はグラフィックは使用できないがケイ線が使える。12行時はグラフィックY軸方向は192か384, 25行なら200か400ドット。なお, X1では行数は25に固定。
CONSOLE〔開始行〕〔,〔行数〕〔, ファンクションキー表示〕〕〕〔,〔カラーモード〕〕〕 ※ファンクションキー表示はNと同じ。 カラーモードは0でカラー, 1でモノクロ。モノクロ時は緑で表示されるが速度が速くなる。これは, FM-7がグラフィックRAMに文字を表示していることを少しでもカバーしようというもの。	CONSOLE 開始行, 行数〔, 開始桁, 桁数〕	CONSOLE 開始行, 行数〔, 開始桁, 桁数〕

8.2 INIT RSMODE RSO RSI WIDTH CONSOLE

BASIC-M25（MZ-2500）	BASIC-S25（MZ-2500）	S-BASIC(MZ-700/1500) ①MZ-700 テープ版(付属) ②MZ-700 QD・Disk版 ③MZ-1500QD・Disk版 にのみ存在
INIT "MEM：〔大きさ〕〔，スプーラの大きさ〕"	INIT"MEM：〔大きさ〕〔，スプーラの大きさ〕"	INIT "RAM：〔大きさ〕〔，スプーラの大きさ〕" ③
INIT"COM番号：〔ボーレイト〔，パリティ〔ビット〔ストップビット〔制御方法〔カナ表現〔削除受信〔送信改行指定〔受信改行指定〔日本語表現〔エンドコード〕〕〕〕〕〕〕〕〕〕〕" ※ボーレイト：75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200のいずれか (turboは0～6だがMZ-2500ではボーレイトそのものを指定。省略時9600) パリティ：E(偶数)Nまたは省略(なし)O(奇数)のいずれか ビット：7または8。省略すると8 ストップビット：1(1ビット)2(1.5ビット)3または省略(2ビット)のいずれか 制御方法：X(XON/XOFF制御する)R(RTSでXON/XOFFする)Nまたは省略(しない)のいずれか 削除受信：DELコード(7F$_H$)受信時の処理 N(NULLつまり00$_H$に変換) B(BSつまり08$_H$に変換) Dまたは省略(変換しない)のいずれか 改行指定：C(CRのみ) Lまたは省略(CR+LF)のいずれか 日本語表現：J(PC-9801コンパチの1B$_H$+4B$_H$/1B$_H$+48$_H$) M(MZ-5500/6500コンパチの1B$_H$+09$_H$/1B$_H$+01$_H$) P(旧JISコンパチの1B$_H$+24$_H$+40$_H$/1B$_H$+28$_H$+48$_H$) Nまたは省略(シフトJIS)のいずれか エンドコード：キャラクタコード00$_H$～1F$_H$のいずれかに40$_H$か60$_H$を加えた文字。Dまたはdなら04$_H$ファイルの終了コードになる	INIT"COM番号：〔ボーレイト〔，パリティ〔ビット〔ストップビット〔制御方式〔カナ表現〔削除受信〔送信改行指定〔受信改行指定〔日本語表示〔エンドコード〕〕〕〕〕〕〕〕〕〕〕" ※M25と同内容。	INIT "RS番号：モニタリングコード，初期設定コード" ③ ※モニタリングコード：0～255の値で，1のビットの信号をモニタする。 ビット7…全キャラクタ送出 ビット6…送信RTS ビット5～3…使用しない(常に0を指定) ビット2…送信CTS ビット1…送信DCD ビット0…受信DCD 初期設定コード：0～255の値でビット毎に意味を持つ。 ビット7…0＝7ビット，1＝8ビット(データ長) ビット6～4…使用しない(常に0を指定) ビット3・2…00＝未使用 01＝ストップビットは1ビット 10＝ストップビットは1.5ビット 11＝ストップビットは2ビット ビット1・0…00＝パリティなし 01＝奇数パリティ 10＝パリティなし 11＝偶数パリティ
(PRINT ＃ 等を使用)	(PRINT ＃ を使用)	(PRINT ＃ を使用)
(INPUT ＃ 等を使用)	(INPUT ＃ 等を使用)	(INPUT ＃ 等を使用)
WIDTH〔桁数〕〔，行数〕 ※桁数は40か80。行数は12, 20, 25のいずれか。グラフィック非連動。	WIDTH〔桁数〕〔，行数〕 ※M25と同内容。	(常に40桁×25行)
CONSOLE 開始行，行数〔，開始桁，桁数〕	CONSOLE 開始行，行数〔，開始行，桁数〕	CONSOLE 開始行，行数〔，開始桁，桁数〕

SHARP BASIC (MZ-80K/C/1200/80B/2000/2200) ⓪SP-5030 ③MZ-1Z001 ⑤MZ-2Z001 ①SB-5520 ④MZ-1Z002 ⑥MZ-2Z021 ②SB-6520 にのみ存在	備考	働き	分類
	RAMディスクとは余ったメモリをディスクのように使えるもの。X1/turboはOPTION SCREEN 2のときに48K分使える(turboでは2倍まで可)。MZ-1500, 2500はKバイト単位の大きさで可変長。ただしMZ-1500はプリンタスプーラ(バッファ)と合わせて64Kまで、MZ-2500はRAMディスクは128K, プリンタスプーラは64Kまで可。MZ-1500ではRAMカード(オプション)要。	RAMディスクを初期化する。	ファイル操作ステートメント
RSMODE {A/B} 〔, R受信ビット〔, T送信ビット〔, Mモード〕〕〕, RX許可 ②⑤⑥ ※ビット：5, 6, 7, 8のいずれか(初期値8) モード：69～71…ストップビットは1ビット 73～75…ストップビットは1.5ビット 77～79…ストップビットは2ビット 69, 73, 77…奇数パリティ 70, 74, 78…パリティなし 71, 75, 79…偶数パリティ 69～79の中で両条件に合致するものを指定する(初期値79) 許可：0…受信許可 1…受信禁止	INIT等により決めるものと、OPEN時に決めるものとがある。OPENによるものは、RS-232Cアクセス時の文法を示した。	RS-232Cのパラメータを設定する。	
RSO {A/B} データ	ファイルディスクリプタの考え方がないもののための文。N88, F, Hu, M25, S25ではディスク・カセット等と共通のPRINT#(あればWRITE#も)が使える。	RS-232Cにデータを送信する。	
RSI {A/B} 文字変数名	ファイルディスクリプタの考え方がないもののための文。N88, F, Hu, M25, S25ではディスク・カセット等と共通のINPUT#(あればLINE INPUT#も)が使える。	RS-232Cからデータを受信する。	
CONSOLE C桁数 ①～⑥ ※桁数は40か80。行数は常に25。 グラフィックとは非連動。	画面モードを設定する文は各パソコン毎に違い、また、きちんと対応する文が存在しない。つまり、各文毎に移植することはできず、いくつかの文をまとめて別のいくつかの文に移植するということになる。移植の際の重要なキーポイントとなるわけだが、大切なことは、原プログラムで何をしようとしているかをきちんと理解することと、移植したいパソコンにそれができるかどうか、そしてできないならどう対応するかをきちんと考えてから移植することである。WIDTHなら、80桁のプログラムを40桁専用のパソコンに移植するのは難しい。しかし不可能ではない。画面のレイアウトを考え、不要なスペースを省くとか、思い切って左半分または右半分のみの表示にするとか、あるいは左半分と右半分とをキーを押すことで切り替えて表示するとか、原プログラム通りのことができなくとも原プログラムの目的としていることなら実現できるものである(パソコンのワープロは漢字1000文字表示も可能だが、たった8文字で頑張っているハンディワープロだってあるではないか！)。	文字画面のモードを設定する(主に表示桁数)。	画面モード・文字画面制御
CONSOLE S開始行, 終了行 ①～⑥	N, N88, F, SHARPは縦方向だけだが、横方向もできるものもある。またSHARPでは行数でなく終了行になっている点にも注意。N, N88のCONSOLE 10, 5, 0, 1なら、CONSOLE 10, 5(SHARP以外)やCONSOLE S10, 14(SHARP)にする。ただしCONSOLEの第4パラメータが0だったか1だったかは重要なので、きちんと憶えておくようにする。また、N等はウィンドウの外にLOCATEでカーソルを移動できるがHu等ではできない。このときはCONSOLEを一時的に解除するようにする。また、Hu等の横範囲は、SHARPでは取りあえず削除かREMにしておく。	文字画面のモードを設定する(主にスクロール範囲)。	

Multiplanの基礎知識

Masuda Toru
増田 亨

Multiplanはコンピュータで情報を処理するとはこういうことなのかと実感させてくれる本格派だ。そのためにも豊富な機能をいかに使いこなすかが重要なポイントとなってくる。これから3回にわたってMultiplanの内容紹介とともに，いかに使いこなすかを皆さんといっしょに考えてみたいと思う。

パソコンを実務に活用したいと願う人は多いが，実際に活用されているケースは少ない。原因は「良いソフト」がないことに尽きるだろう。しかし，これから紹介するMultiplanは，まさにその「良いソフト」である。

Multiplanを使えば，表を作って電卓で必死に計算するという仕事がパソコンで楽々とできるようになる。集計用紙に定規で線を引いたり，縦横の集計を何回も電卓で計算し直すという作業から開放されるのだ。

Multiplanの機能は作表機能と自動計算機能にわけることができる。

作表機能は，集計用紙に定規で線を引くという作表作業がパソコンでできる機能である。Multiplanを使うメリットは，線の引き直しや項目の追加・削除という変更が手軽にできるようになることである。また小数点以下の桁数とか，数字の前の¥マークの有無という表示形式の変更も命令ひとつで簡単にできる。

自動計算機能は，計算手順をパソコンに覚えさせて自動計算させる機能である。パソコンに覚えさせるといってもBASICでプログラムを組むよりはるかに簡単にできる。画面に表示された表を見ながら視覚的に計算手順を入力していくことができるからである。Multiplanを使うメリットは，縦横集計というような定型的な計算手順が数回の命令で簡単に指定できることだ。また，計算手順をパソコンに覚えさせれば，データが変わった場合の再計算が瞬時にできるようになる。

Multiplanがあれば，集計用紙と電卓を使うタイプの仕事は，すべてパソコンで効率良くできるようになる。

ただ，Multiplanは機能が豊富で奥が深いので，最初に使い方を間違って覚えると，その凄さを十分に利用できない可能性がある。

そこで，Multiplanの基礎(第1回)，機能の詳細と具体例(第2回)，高級テクニックと応用例(第3回)という3段階にわけ，連載でMultiplanを紹介していく予定である。

今回はMultiplanの基礎ということで機能の詳細よりは入門的な内容を中心に紹介する。実務にすぐ使いたいという人も，ひとまずおつき合い頂きたい。使いこなす上で必ず役に立つ内容ばかりである。

では，ユダヤの知恵者の話からMultiplan入門を始めることにしよう。

ユダヤの知恵者の話

どんな悩みでも解決してくれると評判になったユダヤの知恵者がいた。彼に相談すると，誰もの悩みが解決されるというのだ。

ある人が「いったいどうすればどんな悩みでも解決できるのですか？　良い方法があるのなら教えてください」と知恵者に頼み込んだ。知恵者は「私のところにくる人はみんな，どうすればよいかと質問するのです。そこで私は，紙とペンを用意してこう言います。どんな解決方法があるのか，一見不可能な方法でもとにかく思いつく限りこの紙に書いてみてください。ただし具体的に書くこと。自殺するとか，駆け落ちするとか書くだけではだめで，自殺ならどんな方法で自殺するのか，駆け落ちならいつどこにどんな乗物で逃げだすのかまで書かなければだめですよ，とね」

「そこで初めて，どの方法が一番良いかを教えてあげるのですか？」と聞き返すと，知恵者は「とんでもない。相談にやって来た人が，この方法が自分には一番いいと思うのでやってみます，と自分で選んで帰っていくのです」と笑って答えた。

ユダヤの知恵者の話を最初にしたのは，Multiplanが何のためのソフトかを理解するのに助けになる話だからである。

知恵者は，①自分で，②思いつく限りの方法を，③できるだけ具体的に書くことを指示した。そうすると，相談に来た人は最善と思う方法を自分で見つけていくというのである。

これがどういうことかと言うと，相談に来る前までは頭の中だけで考えているから，いつまでたっても具体的な解決策は出てこない。考える内容も，これからどうしようという前向きなことよりも，どうしてこんなことになってしまったのかという後ろ向きなことが中心になってしまう。これではいつまでたっても解決策は出てこない。

しかし，ユダヤの知恵者の教えのように，これからのことだけに集中すれば，いくつかの方法は思い浮かぶはずだ。ただ，それも抽象的な表現だけでは，やはり「解決策」にはならない。だから，知恵者はできるだけ具体的にと教えるのである。

具体的に方法を書いてみれば，実現性や

図1　どうやったら儲かるか

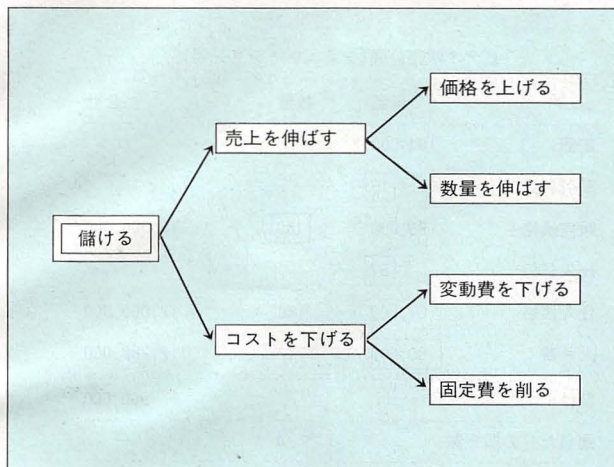



図2　儲けの試算

項目	金額	計算
商品正価	100,000円	
販売価格	80,000円	(10万円×80%)
販売数量	150台	(3台/週×50週)
売上合計	12,000,000円	(8万円×150台)
仕入れ価格	70,000円	(10万円×70%)
仕入れ額計	10,500,000円	(7万円×150台)
人件費	2,400,000円	(20万円×12カ月)
儲け	−900,000円	(売上−仕入れ−人件費)

図3　儲けを100万円にする

①数を増やす

150台	90万円の赤字
200台	40万円の赤字
300台	60万円の儲け
400台	160万円の儲け
340台	100万円の儲け

②価格を上げる

20%引き	90万円の赤字
15%引き	15万円の赤字
10%引き	60万円の儲け
7%引き	105万円の儲け

③仕入れ価格を下げる

70%	90万円の赤字
65%	15万円の赤字
60%	60万円の儲け
57%	105万円の儲け

④固定費を削る

月20万円	90万円の赤字
月15万円	30万円の赤字
月10万円	30万円の儲け
月5万円	90万円の儲け
月4万円	102万円の儲け

損得などを比較しやすくなる。だから自然に解決方法が見つかる仕組みなのだ。

Multiplanは，この知恵者の教えをビジネス分野で実践するために作られた道具なのである。

ビジネスへの応用

人はそれぞれ悩みを抱えて生きている。しかし，ビジネスに限定すれば悩みは「どうやったらもっと儲かるか」に尽きるはずだ。人生は金がすべてではないが，ビジネスでは金儲けがすべてである。別に金にガツガツしろとか，金に汚くなれというのではない。一定のルールやフェアプレイ精神が要求されるのはゲームやスポーツと同じである。ゲームやスポーツは「勝つ」という単純な目標の上に成り立っているのに対し，ビジネスは「儲ける」という単純な目標の上に成り立っているのである。

儲けるには売上を伸ばし，コストを削ることである。売上を伸ばすには販売価格を上げる，販売数量を増やすという2つの方法がある。コストダウンは，変動費を削るか固定費を削るかの2つに分かれる(図1参照)。

数量・単価・変動費・固定費の4つの数字を変化させることで儲けを増やすことができるわけだ。

しかし販売量を増やすとか，固定費を削るというのは「具体的に」という知恵者の教えに反している。

売上を伸ばすには，まず具体的に目標を決めること。このときに重要なのは個々の数字が実現可能な数字かどうかにはこだわらず，まず具体的に数量・単価・変動費・固定費を数字で表して儲けを試算してみることである。

仮に1年間で100万円儲けたいとする。方法は正価が10万円の商品(たとえばビデオ)を1人で売りさばくとする。

具体的に儲けを計算してみよう（図2参照)。

正価では売れないだろうから，20%引きで売るとする。台数は週に3台，年間で150台と考えてみる。

仕入れ価格は正価の70%とする。これは売れた台数に比例する費用だから，変動費である。そのほかに人件費として月20万円を見込む。これは売れても売れなくてもかかる費用だから，固定費である。

このような条件で試算してみると，儲けは図2のように90万円の赤字になる。

目標は年間で100万円儲けることである。どうすれば90万円の赤字を100万円の儲けにすることができるのだろうか。ユダヤの知恵者の教えに従えば，思いつく限りの具体的な方法を書いてみろということになる。

儲けは，図1のように数量を増やす，価格を上げる，変動費(仕入れ単価)を下げる，固定費(人件費)を削るという方法で増やすことができる。

まずひとつだけの要素を変化させて，他の3つの要素が最初の条件どおりであった場合に目標額「100万円の儲け」を達成するための方法を考える。

図3のように，数だけであれば150台を340台に増やすこと。売値だけを変えるとしたら，20%引きを7%引きにする。仕入れ価格を変えるとしたら70%を57%にする。固定費を削る場合は月20万円を月4万円にすれば儲けは100万円になる。

実現の可能性はどれも低そうである。ビジネスは相手があって成り立つのでそうそううまくいくはずがない。たとえば，340台を1年間で売るには毎日休みなく1台売る

図4　Multiplanによる試算(1)

ビデオ販売計画(シミュレーション用)

	単価	数量	合計
正価	100,000		
割引(%)	20		
販売価格	80,000	150	12,000,000
仕入(%)	70		
仕入価格	70,000	150	10,500,000
人件費	200,000	12	2,400,000
儲け			−900,000
週当たりの販売量		3	

図5　Multiplanによる試算(2)

ビデオ販売計画(シミュレーション用)

	単価	数量	合計
正価	100,000		
割引(%)	15		
販売価格	85,000	180	15,300,000
仕入(%)	67		
仕入価格	67,000	180	12,060,000
人件費	190,000	12	2,280,000
儲け			960,000
週当たりの販売量		4	

ということである。これが大変な数字であるということはセールスの経験がある人にはすぐわかることだろう。しかし，台数をもう少し減らして，他の要素を変えて100万円の儲けを達成する方法はあるはずだ。

4つの要素を少しずつ変化させて，その結果100万円儲かるような組み合わせにすればもっと現実的な計画になる。

いよいよMultiplan登場

今までやってきたような単純な試算だけでも，電卓でやっていたのでは時間ばかりかかってしまう。しかし，Multiplanであればいままでのような試算は短時間でできるし，これから説明するような4つの数をいろいろな組み合わせで変化させて結果を出すことが実に簡単にできるようになるのである。

図4がMultiplanで作ってみた表である。画面上には出てこないが，Multiplanで表をつくるときは実際の計算方法も指定してパソコンに覚えさせる。

写真1　分割画面による表の対照

写真2　IRR関数による損益処理

写真3　販売計画のシミュレート

この表では数字をひとつ変えると，パソコンが瞬時に計算し結果をはじき出してくれる。たとえば台数を変える，つまり100を340に変えると，その瞬間に儲けが，−900,000から1,000,000に変化する。

だから台数，販売価格(%)，仕入れ価格(%)，人件費をいろいろ変えると変える度に儲けの試算結果がすぐ表示される。儲けを100万円にするための台数，販売価格，仕入れ価格，人件費を数字を変えて試算しながら決められるわけだ。

この作業は専門的な言葉で「会話型でシミュレーションを行う」と言う。ようするに，やり方(計算方法)を決めてパソコンに覚えこませておいて，あとは人間が数を与えてやる度にパソコンが計算結果を答えるというわけだ。Multiplanというのは，こういう作業をするための便利な道具なのである。

実際にMultiplanで作った表で，試算結果が即座に表示されるということの凄さは，文章や言葉で表現しても，うまく伝わらない。自分でパソコン上でやってみると，まざまざとパソコンの威力が実感できるのだが……。

この会話型のシミュレーションをした結果のひとつが図5である。つまり台数180台，販売価格15%引，仕入れ価格67%，人件費19万円であれば，儲けが96万円になるのである。ただし，答えはこの1種類だけではない。

台数をもう少し減らして，他の要素を変えて儲けを100万円にする計画は他にも作成できる。何種類もの具体的な目標を作って

みるというユダヤの知恵者の教えがこれにあたるわけだ。

Multiplanでシミュレーションをするには，先を読む力や，実際にはどうなりそうだという経験が重要になってくる。同時にいろいろな可能性を追求してみるという柔軟性も必要になる。

パソコンでMultiplanを使えば，新人でもベテランと同じ仕事ができるようになるわけではない。むしろ，先を読む力や経験の差が，手作業のときよりもはっきりと出てくるようになってしまう。もっとも，可能性の追求という柔軟性の面では，新人のほうがベテランよりも勝っているかもしれないが。

もうひとつの使い方

Multiplanは数字をいろいろ変えて結果を調べるというシミュレーション作業ではなく，1回だけの作表・計算作業にも有効である。集計表をきれいに早く作りたいという場合もMultiplanは便利な道具になる。

ただし，この場合は手作業よりもMultiplanでやるほうが楽になるのは，Multiplanをかなり使いこなせるようになってからである。

パソコンの初心者が，Multiplanを使ってもそう簡単にはきれいな表は作成できるわけではない。しかし，Multiplanが自在に使いこなせるようになるとほとんどの作表作業は，手でやるよりずっと速く，簡単にできるようになる。

これに対し最初に説明したシミュレーション作業は，電卓での手作業には限界があるが，Multiplanを使えば初心者でも短時間に高度なシミュレーションができるようになるのである。だから，最初はシミュレーション作業に使いながら，慣れてきたら作表作業に使うというような使い方が良いだろう。

Multiplanの基本

Multiplanは，図6のように集計用紙とほぼ同じだと考えればよい。ひとつひとつの□の中には数字，文字，計算式を書くことができる。実際の集計用紙との一番の違いは計算式を書くということである。

図6のAとBの合計の結果をCに入れたい場合にはCの位置にA＋Bというような

図6　Multiplanは集計用紙と同じ

			A	B	C	D		
			A'	B'	C'	E		
						F		
						G		
						H		
						I		

式を書く。そうするとCの位置には計算結果が表示される（計算式は普段は画面には表示されない）。

だからAやBの場所の数字を変更すれば常にその計算結果がCの位置に即座に表示される。

A＋Bという計算式はMultiplanでは，

R2C4＋R2C5

という書き方をする。R2C4というのは縦が2番目で横が4番目の□という意味で図6のAの位置を指しているわけだ。

この書き方はかなり面倒だが，MultiplanではAの場所にカーソルをもっていってリターンキーを押せば自動的にR2C4と入力行に表示され，＋と書いてからBの位置にカーソルを移動してリターンキーを押せばR2C5と表示される。このカーソル移動キーで場所を指定できるという機能があるので場所の指定は簡単にできる。

実際に計算式を書く場合に非常に便利なのが，関数機能と複写機能である。

①関数機能

関数機能は，よく使う計算のパターンは簡単に指定できる機能である。代表的なものには合計関数がある。

表形式の計算では，図6のDからHまでの合計結果をIの場所に表示するというような計算が多い。

このときDからHまで□をひとつひとつ指定しながらたし算を指定するのは面倒である。

こういうときは，

SUM（R2C7：R6C7）

D　：　H

という式をIの場所に書き込めばすむようになっている。この場合もR2C7という場所の指定はカーソル移動キーで簡単にできる。

関数機能は，合計関数以外にも豊富に用意されている。関数機能が自在に使いこなせると，計算方法の指定作業がかなり簡単になる。

関数機能については次回で詳しく紹介することにしたい。

②複写機能

表形式の計算指定で，もうひとつよく出てくるのが，まったく同じ計算をあちこちでやる必要がある場合である。

CALCソフトの歴史

Multiplanに似たソフトはたくさんあって，CALCソフトとかスプレッドシートと呼ばれる。

この種のソフトは，アメリカで爆発的にヒットしたVisicalcをもとに改良され発展してきたものである。

Visicalcは，ハーバード・ビジネススクールに在学中のD.S.ブルックリンが，経営計画・経営分析の計算実習のあまりの大変さに，なんとかパソコンを利用して楽をしようと考えたのがきっかけで開発された。パソコンに計算手順を覚えさせて自動計算させるというのが基本機能で，BASICでプログラムを組むよりも簡単という意味で，簡易言語とも呼ばれた。

しかしVisicalcは，他のソフトとデータの互換性がない，作表機能が貧弱，一度に作れる表の大きさが小さいなどの弱点があり，各社から改良版が次々に発表された。中でもCP/M上で動くSuper Calcは8ビットパソコン用のCALCソフトとして非常に完成度が高いものである。

16ビットパソコンが一般的になると，その高機能をいかすためのCALCソフトが登場してくる。代表的なのがここで紹介したMultiplanである。16ビットマシンの代表的実務ソフトが，X1 turboで使える意義は大きい。

Multiplanは，命令の選択方法や計算の指定方法が簡単になり，一方でファイル間のデータ参照という高級機能も用意されていて，応用範囲が広くなっている。Multiplanの日本語対応版は，日本のCALCソフトの標準といって良いものである。

図7　作業の選択肢

選択肢	作業内容
A／文字	文字を集計用紙に書き込む
B／空白	□を空白にする(消す)
C／複写	□の内容の他の□に複写する
D／削除	行や項目を表から削除する
E／修正	□の内容を書き換える
F／書式	表全体や□ごとに書式を指定する
G／セル移動	□の内容を他の□に移動する
H／解説	画面上の各機能の説明を表示する
I／挿入	行や項目を表に追加する
L／保護	□の内容を変更できないようにする
M／移動	行や項目を他の場所に移動する
N／名前	□に名前を付ける
O／オプション	自動再計算機能，警告音機能などのオン／オフ
P／印刷	プリンタやフロッピーへの出力
Q／終了	終了する
S／並替え	表を行単位で並びかえる
T／転送	他のプログラムとのデータのやりとりをする
V／数式	数字や計算式を□に書き込む
W／窓	画面を分割して，表の2個所を同時に表示する
X／連結	複数のファイルをひとつのファイルにする

図8　6つの基本の選択肢

選択肢	作業内容
A／文字	文字を集計用紙に書き込む
B／空白	□を空白にする(消す)
C／複写	□の内容の他の□に複写する
P／印刷	プリンタやフロッピーへの出力
Q／終了	終了する
V／数式	数字や計算式を□に書き込む

図6で言えば，CにはA＋Bの結果を，C'にはA'＋B'の結果を表示したいというように，同じ計算をやれば良い場合の計算方法の指定に便利なのが，複写機能である。

CをC'の位置に複写してやると，MultiplanではC'にはA＋Bの結果ではなくA'＋B'の結果が表示される。つまり自動的にAをA'に，BをB'に変更して複写してくれるのである。

複写元や複写先の位置指定もカーソル移動キーで指定できる。

この複写機能をうまく使えば，表を作成する時間が大幅に短縮できる便利な機能である。

Multiplanでの作業

Multiplanでは，集計表に文字・数字・計算式を書くという作業以外にも，図7のようないろいろな作業ができる。作業の指示は，画面の下に図7の選択肢が表示されるので，その選択肢をスペースキーとバックスペース(BS)キーを使って選ぶという方法で作業指示をする。

書式設定という作業を選ぶと，いろいろと追加で指定したい内容の選択肢が表示されるようになっていて，かなり複雑な作業が，簡単に指示できるようになっている。

ただし，Multiplanは最初に書いたように機能が豊富な奥の深いソフトであり，それが初心者にとっては大きな欠点でもある。まだ使いこなせないような機能まで選択肢として表示されるので，適切な作業を選ぶのが大変になってしまうのである。

だから，Multiplanを使い始めるときには，とにかく単純な使い方を心掛けることだ。

図7のようないろいろな機能を全部覚えようとするのではなく，まず図8のような6つの作業だけでMultiplanを使うことである。これだけの機能が使えれば，まず一通りのことはできる。

注意したいのが「見てくれに凝るな」ということだ。

初心者がMultiplanをすぐ役に立てられるのは，シミュレーション作業（試算の繰り返し）をやる場合である。これは計算方法さえ指定することができれば，あとは初心者でも存分にシミュレーション作業をすることができる。

この場合，見出し，表示の書式（3桁ごとにカンマをつけるとか，□の幅を変える）というような作業は最初はやろうとしないことである。

シミュレーション作業は計算結果が大切なのであって，きれいなとか，見やすいというのは2次的なことである。最初からそういう2次的なことにこだわっていては，Multiplanは使いにくいと誤解してしまう。まず最低限必要な作業に限ってMultiplanを使うべきだ。

表をきれいに仕上げるためには，最初は操作説明書をあちこちひっくり返すことになる。そうやっているうちに「Multiplanはなんて面倒なんだ。これなら手でやるか，ワープロで表を書いたほうがずっといい」ということになりかねない。これは使い方が基本的に間違っているのである。図8のような必要最小限の機能を使うだけでもかなり複雑なシミュレーションができるのである。

シミュレーションをするのに重要なのはMultiplanの特殊な機能ではなくて，どういう計算方法を指定したらシミュレーションができるかという，シミュレーション・モデル(試算方法)を考える力である。先を読む力，数字の妥当性を判断できる経験，可能性を追求する柔軟性である。

Multiplanが使いこなせるようになってくると別にシミュレーション作業のようなものでなくても，Multiplanは便利な道具になる。

書式設定や印刷書式の設定という機能が自由自在に使いこなせるようになると，手で表を作るよりきれいな表がごく簡単にできるようになる。

特に，作っている最中にあちこち変更することが簡単にできるので，手作業のように一部を変更するために最初からやりなおしというようなことはなくなる。

あとから，項目の幅を変えたいというようなことは誰でも経験があるだろう。Multiplanではこういう変更作業がごく簡単にできるのである。また，9000と入力すれば￥9,000というように表示するようにするような機能も準備されている。

これらの機能は次回以降で取り上げるこ

図9　表は分割して作ること

個々の表を1画面におさまるようにすれば，作業が楽だし，わかりやすい表になる。
1画面におさまらない時は，小計を計算する表をいくつか作って，総括表は必ず1画面におさまるようにする。

とにする。もう一度強調したいのは，最初からそういう機能ばかりに凝ってしまうと，道に迷ってしまうということだ。

Multiplanを使い始めたばかりなのに，こういう作業をすぐにやろうとしても，説明書をあちこちひっくり返す手間ばかりで，手作業より効率が落ちる。そういう期間をしばらく我慢してMultiplanに慣れてしまえば，逆に手作業が面倒に思えることは確実である。

表の作り方

Multiplanを自在に使いこなせるようになると，大きくて複雑な表が作れるようになる。Multiplanは，最大で縦255行，横の項目数63という表が作れることになっている。

しかし，だからといって横の項目数が40も50もあるような表を作ることは意味がない。

画面の大きさ以上の表を作っても一度には全体を見れない。一度に全体が見れない表は作るのも大変だし，実際に試算したりするにも使いにくい。

では，なぜ画面の大きさ以上の表が作れるようになっているかというと，それは図9のような使い方をするためである。

最初に説明した100万円儲けるためのシミュレーション作業には，実際にはもっと多くの要素が関係する。たとえば商品の運送費，運転資金を借り入れた場合の利息，通信費，倉庫費用など数え上げればいくらでも出てくる。

だからと言って，大きな表ばかりを作ってもわかりにくくなるだけである。最初のシミュレーションは，要素を絞ってあるから試算作業がしやすいし，内容が理解しやすいのである。

だから，実務的なシミュレーションをしたい場合でも基本的には最初の例のような大きさの表でやるのでなければ意味がない。ただし，仕入れ費用を変動費，人件費を固定費というように捕らえてシミュレーションをするのである。

そして変動費いくら，固定費いくらという目標が決まったら，今度はもう少し細かく変動費の中身，固定費の中身を試算してみる。

逆に集計作業をしていくような場合も，明細表をいくつか作って小計表を作り，その小計を集めて合計するための総括表を作る。

それぞれの表は必ず画面に入りきるように作っておけば，常に関係項目がすべて一度に見えているわけだから，ポイントがつかみやすい表になる。

大きな表をただ作ってみても，いったい何がポイントなのかはよくわからなくなってしまう。

そのために，ここで説明したような表の作り方をしておけば，各レベルごとにポイントが明確になる。これはMultiplanを使う上でだけではなく，資料を作る時にすべてに使える考え方なのである。

最後に

今回は入門編であり，実務にすぐ使ってみたいと思っている人には，少し情報不足だったかと思う。しかし，Multiplanを使いこなす基礎としては必要な内容なのである。

次回以降で，実務ですぐ使えそうな表を取り上げながら，Multiplanの各機能を詳しく紹介していく予定である。

実務にパソコンを活用したいと考えている人で，困っていることや質問があれば，ハガキに書いて送ってほしい。できる限りこの連載のなかで取り上げ，いっしょに考えていきたい。

また，読者の皆さんの中で，実務にパソコン（特にMultiplan）を活用しているという人がいたら，連載中に紹介したいと考えているので，その活用方法などもぜひ送ってください。

CALCソフトとデータベースソフト

実務用のソフトには，MultiplanのようなCALC系の他にデータベースがある。両者は機能や用途がまるで違うソフトなので注意が必要である。

CALC系のソフトは，マクロ機能によって表形式の計算，結果の表示・印刷をする「特定目的用のプログラム言語」なのである。

このデータとあのデータの合計をある場所に表示(印刷)しろ，何行目に横線を引けというような命令を組み合わせて，表形式の計算・結果の表示をするための言語で，BASICに比べプログラムの作成や変更が非常に簡単なことから，「簡易言語」とも呼ばれる。

データベースは，BASICでいえば「DATA文だけ」を作るソフトである。そのDATA文を使って何をやるかという処理部分はユーザーが独自にプログラムを作成する。もっとも最近のデータベースはプログラム言語や自動実行機能を持っており，分類・集計・検索という頻繁に使う機能はひとつの命令でできるようになっている。

両者とも表形式でデータを並べるので，同じ種類のソフトという誤解が多い。簡単に言うと，CALC系ソフトは数値処理を主としているが，データベースタイプのアプリケーションは，文字列の処理が中心となるので，情報の種類によって使い分けたほうがよい。しかし，計算をしろ，線を引けという命令が多く，表処理が複雑な場合にはCALC系ソフトが向いている。

逆に処理は単純だが，データ量が多く他のデータと重複する部分が多い場合などはデータベース向きなのである。

表集計の例(CALCソフトの用途)

第3期　販売実績　(千円)

	10月	11月	12月	計
テレビ	480	288	864	1632
ラジオ	12	6	15	33
ワープロ	344	172	774	1290
計	836	466	1653	2955

試験に出るX1

第8回

FDCは挑戦的である

Iwai Ippei
祝 一平

私が講師の祝一平だーぞと。

今月は先月に引き続いて，フロッピーディスクについてやるのである。先月は概論および，HuBASIC上におけるソフト的なフォーマットであった。今月の方針としては，ひとまずFDCに詰め寄り，抜く手も見せずにフロッピーディスクの深遠へと向かうつもりである。残念ながら，フォーマットなどのエグイ部分は来月回しである。

とりあえずFDCについて

FDC (Floppy Disk Controler) とは，そのものずばりに，フロッピーディスクをコントロールするためのLSIである。FDCの種類は，「単密度専用」と「倍密／単密両用」の2つに分かれるが，「単密度専用FDC」はすでに過去の遺物なので，「倍密／単密両用FDC」についてのみ書くことにする。

倍密／単密両用FDCは，

1）FD1791ファミリ
2）μPD765ファミリ
3）その他

に分かれる。

表1　MB8876/8877のコマンド

タイプ		コマンド名称	動　作	7 6 5 4 3 2 1 0	X1での標準的な値
I	0	リストア	トラック0へ，ヘッドを移動する	0 0 0 0 h V r_1 r_0	02_H＝0 0 0 0 0 0 1 0
	1	シーク	所定のトラックへ，ヘッドを移動する	0 0 0 1 h V r_1 r_0	$1E_H$＝0 0 0 1 1 1 1 0
	2	ステップ	ヘッドを1トラック移動する	0 0 1 u h V r_1 r_0	$3A_H$＝0 0 1 1 1 0 1 0
	3	ステップ・イン	ヘッドを1トラック内側へ移動する	0 1 0 u h V r_1 r_0	$5A_H$＝0 1 0 1 1 0 1 0
	4	ステップ・アウト	ヘッドを1トラック外側へ移動する	0 1 1 u h V r_1 r_0	$7A_H$＝0 1 1 1 1 0 1 0
II	5	リードデータ	ディスクのデータ(データフィールド)を読む	1 0 0 m S E C 0	80_H＝1 0 0 0 0 0 0 0
	6	ライトデータ	ディスク(データフィールド)へデータを書き込む	1 0 1 m S E C a_0	$A0_H$＝1 0 1 0 0 0 0 0
III	7	リードアドレス	ディスクのIDフィールドを読む	1 1 0 0 0 E 0 0	$C0_H$＝1 1 0 0 0 0 0 0
	8	リードトラック	ディスクの1トラック分の全データを読む	1 1 1 0 0 E 0 0	$E0_H$＝1 1 1 0 0 0 0 0
	9	ライトトラック	ディスクへ1トラック分の全データを書き込む	1 1 1 1 0 E 0 0	$F0_H$＝1 1 1 1 0 0 0 0
IV	10	フォースインタラプト	割り込みを発生させる	1 1 0 1 I_3 I_2 I_1 I_0	$D0_H$＝1 1 0 1 0 0 0 0

表2-1（TYPE Iフラグ機能表）

フラグ	機　能
u	トラックレジスタ更新フラグ（u）は，ヘッド移動に際し，トラックレジスタを更新する事を指示する。 u＝1：トラックレジスタを更新する。 u＝0：トラックレジスタを更新しない。
h	ヘッドロードフラグ（h）は，コマンド実行開始時にヘッドをロードするか，ヘッドをメディアから離すかを指示する。 h＝1：コマンド実行開始時にヘッドロードする。 h＝0：コマンド実行開始時にヘッドを離す。
V	トラック照合フラグ（V）は，ヘッド移動後，ディスクのトラック番号とトラックレジスタの照合を行うかを指示する。 V＝1：トラックの照合を行う。 V＝0：トラックの照合は行わない。
r_1, r_0	ステップレートフラグ（r_1, r_0）は，ステップパルス出力の間隔を指定する。（下表）

r_1	r_0	クロック 1MHz（X1ノーマル）	クロック 2MHz（turboの2HD）
0	0	6ms	3ms
0	1	12ms	6ms
1	0	20ms	10ms
1	1	30ms	15ms

表2-2（TYPE II，IIIフラグ機能表）

フラグ	機　能
m	マルチレコードフラグ（m）は連続セクタでリード／ライトするかを決める。 m＝1：連続セクタ（セクタ番号が増加する方向）でリード／ライトを行う。 m＝0：単一セクタでリード／ライトを行う。
S	サイドフラグ（S）はサイド番号指定に使う。 S＝1：サイド番号のLSBが1のとき一致したものとみなす。 S＝0：サイド番号のLSBが0のとき一致したものとみなす。 （このフラグは，Cフラグが1のときのみ有効）
E	ディレイフラグ（E）はHLT信号のサンプリングのタイミングを指定する。 E＝1：HLD（ヘッドロード）信号を“H”としたのち15ms待ち，HLT信号をサンプリングする。 E＝0：HLD信号を“H”としたのち，ただちにHLT信号をサンプリングする。
C	サイド番号比較フラグ（C）はサイド番号と比較（チェック）するかどうかを指示する。 C＝1：サイド番号の比較を行う。 C＝0：サイド番号の比較を行わない。
a_0	アドレスマークフラグ（a_0）はデータアドレスマークに何を書くか指定する。 a_0＝0：データアドレスマークに（FB)$_H$（Data Mark）を書く。 a_0＝1：データアドレスマークに（F8)$_H$（Deleted Data Mark）を書く。

表2-3（TYPE IVフラグ機能表）

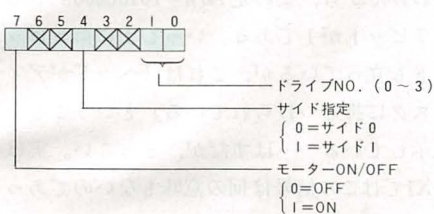
フラグ	機　能
I_0	I_0=1でREDY入力の立ち上りでIRQ発生（IRQ=“H”）。
I_1	I_1=1でREDY入力の立ち下がりでIRQ発生。
I_2	I_2=1でインデックスパルス検出時にIRQ発生
I_3	I_3=1で無条件でただちにIRQ発生

注意：X1/X1 turboに使われているMB8877のIRQ信号はOPEN（どこにもつながっていない）だから，IRQは何の機能も果たさない。

表4（FDC関係I/Oアドレス）

0FF8H	ステータスレジスタ(STR)/コマンドレジスタ(CR)	IN/OUT
0FF9H	トラックレジスタ(TR)	IN/OUT
0FFAH	セクタレジスタ(SCR)	IN/OUT
0FFBH	データレジスタ(DR)	IN/OUT
0FFCH	FM方式指定/ドライブNO，サイド，モーターON	IN/OUT
0FFDH	MFM方式指定	
0FFEH	1.6Mタイプ(2HD)指定	
0FFFH	500K(2D)/1M(2DD)切り替え	

□の部分はturbo用

0FFCH出力内容の意味

FD1791はウエスタン・ディジタル社が開発したFDCで，X1/X1turboに使われているMB8877（富士通製）はFD1791ファミリに属する。それに対してμPD765は日本電気の開発したFDCである。当然のことであるがこれら2つのファミリの間に互換性などという「なごやかな人間関係」は存在しない。

ちなみに，FMシリーズのFDCはX1と同じMB8877，MZシリーズはMB8876を使っている。8877と8876の違いは，バスラインが正論理か，負論理の違いだけである。具体的には，データの0/1が反転するのである。すなわち，X1でディスクに&HFF=&B11111111というデータを書き込み，そのディスクをMZのドライブに差し込むと&H00＝&B00000000が読み出せるというわけである。その他はまったく同じであるから，唯一の問題はX1↔MZ間でディスクのやり取りをする場合だけである。おっと，忘れちゃいけない。X1とMZのディスクの互換性といえば，もうひとつあるのだ。すなわち，SIDE 0/1が逆なのだ。試してみるとわかるが，X1の第0レコードはMZの第16レコードである。逆に，MZの第0レコードはX1の第16レコードである。

本題に入る。表1がMB8876/8877のコマンド，表2が各フラグの意味，表3がステータスレジスタの意味である。どーだ，これだけでは意味がわからんだろう。そこで順に解説していくわけだが，その前に表4がある。これがX1/X1turboでのFDC関係のI/Oアドレスである。青色に塗られた部分はX1 turbo用なので，とりあえず今月は使わない。

表3-1（ステータスレジスタの各ビットの意味）

コマンド ＼ ステータスレジスタ		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
TYPE I	すべてのコマンド	NOT READY	WRITE PROTECT	HEAD ENGAGED	SEEK ERROR	CRC ERROR	TRACK00	INDEX	BUSY
TYPE II	リードデータ	NOT READY	0	RECODE TYPE	RECODE NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
	ライトデータ	NOT REAEY	WRITE PROTECT	WRITE FAULT	RECODE NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
TYPE III	リードアドレス	NOT READY	0	0	RECODE NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
	リードトラック	NOT READY	0	0	0	0	LOST DATA	DATA REQUEST	BUSY
	ライトトラック	NOT READY	WRITE PROTECT	WRITE FAULT	0	0	LOST DATA	DBTA REQUEST	BUSY
TYPE IV	(他のコマンド実行中の場合)	(今まで実行していたコマンドのステータスビットと同様の意味)							0
	(実行中のコマンドのない場合)	NOT READY	WRITE PROTECT	HEAD ENGAGED	0	0	TRACK00	INDEX	0

表3-2（ステータスの意味）

コマンド	ステータス	意味
TYPE I	NOT READY (bit 7)	NOT READY=1でディスクドライブが動作可能状態でない事を示す。
	WRITE PROTECT (bit 6)	WRITE PROTECT=1でディスクへの書き込みが禁止されている事を示す。
	HEAD ENGAGED (bit 5)	HEAD ENGAGED=1でヘッドがメディアに押しつけられている事を示す。
	SEEK ERROR (bit 4)	SEEK ERROR=1でベリファイ動作が成功しなかった事を示す。これは，IDフィールドが検出されなかったか，IDフィールドのトラック番号とトラックレジスタの内容が一致しない事による。
	CRC ERROR (bit 3)	CRC-ERROR=1で，IDフィールド読み出し時に読み出しエラーがあった事を示す。
	TRACK00 (bit 2)	TRACK00=1でディスクのヘッドがトラック0の上にある事を示す。
	INDEX (bit 1)	INDEX=1でインデックスホールを検出した事を示す。
	BUSY (bit 0)	BUSY=1でFDCがコマンド動作中である事を示す。
TYPE II／III	NOT READY (bit 7)	NOT READY=1でディスクドライブが動作可能状態でない事を示す。
	WRITE PROTECT (bit 6)	WRITE PROTECT=1でディスクへの書き込みが禁止されている事を示す。
	RECODE TYPE /WRITE FAULT (bit 5)	リード動作の時，RECODE TYPEの表示として，デリーテッドデータマークの時セットされ，データマーク時リセットされる。ライト動作の時，WRITE FAULTの表示として，書き込み動作が打ち切られた事を示す。
	RECODE NOT FOUND (bit 4)	RECODE NOT FOUND=1で指定されたトラック番号，サイド番号，セクタ番号を持ち，正しく読み出されたIDフィールドがなかった事を示す。
	CRC ERROR (bit 3)	CRC ERROR=1でディスクからIDフィールドもしくはデータフィールドの読み出しにおいて，読み出しエラーがあった事を示す。
	LOST DATA (bit 2)	LOST DATA=1で所要時間内にDRの読み出し，もしくは書き込みが行われなかったデータがあった事を示す。
	DATA REQUEST (bit 1)	DATA REQUEST=1で，FDCがプロセッサに対してDRの読み出しもしくは書き込みを要求している事を示す。
	BUSY (bit 0)	BUSY=1でFDCがコマンド実行中である事を示す。

TYPE Iの実習

FDC，MB8877を実際に使ってみるのである。まず，TYPE Iのコマンドは機械語でなく，BASICからでも使えるのでちょうど良い肩ならしである。ただし，FDCを誤って動かすとディスクの内容がパーになったりすることがしばしばあるので，書き換わってもよいディスク以外はセットしないように。

FDCを使う前にしなければならないのが，ドライブのセレクトとモーターONである。これには，0FFCH番地を使うのである。まずドライブ1（"1:"）に壊れてもよいフォーマットしたディスクを入れて，BASICから直接

OUT &HFFC, &H81

と打ち込んでいただきたい。ドライブ1の赤いLEDが付き，ディスクが回り出したはずである。不幸にしてドライブ0しか持っ

てない人は，＆H81のかわりに＆H80である。ドライブ0が回り出すであろう。冷たいようだが以下同様。

いつまでも回していてもしかたがないので止めることにする。

OUT ＆HFFC, ＆H01(もしくは＆H00)と打ち込むと，約1秒後にディスクが止まる。なぜ1秒後かというと，これはディスクが回り出してから，読み書きできるようになるまでに時間がかかるからなのだ（わかるかな？)。つまり普通のディスクアクセス法として一番自然な方法は，

①ディスクを回し始める。
②ディスクが規定の速度（毎分300回転)になるまで待つ。
③読んだり書いたりする。
④ディスクを止める。

となっている。

問題は，①〜④をやった後で，すぐにまた①〜④に飛んでくるケースが多いことにある。④で一旦ディスクが止まってしまっているから，(すぐさま）次にやってきたときに，またディスクが定常状態になるまで待たなければならないのである。これでは時間の無駄が大きいので，④で「モーターオフ」の信号を受け付けても，すぐにはディスクのモーターを止めず，1秒間の遅延時間を置くのである。これで，次にすぐさまやってきても，待たされないですむのである。めでたし，めでたし。

では，もう少し進んでみる。リスト1である。

まずはFFC$_H$←81$_H$で，ドライブ1のモーターをONする。GOSUB “WNBSY”で，ディスクが正しく定常状態になるまで待つ。次にFF8$_H$（コマンドレジスタ）へ02$_H$（表1のリストアコマンド）をOUTする。これで，ドライブのヘッドは第0トラック（一番外側）へ移動してくれる。もちろん，ヘッドという重さのある「ブツ」が動くわけで多少の時間がかかるから，130行のGOSUB〜で動作の完了を待ってやる。

140行〜170行は「シーク」である。このコマンドの使い方は，まずは下準備として，FFB$_H$（データレジスタ）に目的トラック番号をOUTし，FF9$_H$(トラックレジスタ)へ現在のトラック位置をOUTする。次にコマンドレジスタへシークコマンド（1E$_H$)を送れば，「ククククッ」という音を立てて，第20トラックを目指してヘッドが動いてくれる。

その次はステップ・インで，ようするに1トラックだけ内側にシークする。

さて，220行からのサブルーチンであるが，これはFF8$_H$（ステータスレジスタ）を見張りつつ，MSBとLSB（第7ビットと第0ビット）がともに0になるのを待つものである。第7ビットは「NOT READY」，すなわち「まだ準備できてないよ」という意味。第0ビットは「BUSY」，すなわち「いま忙しいのっ！」もしくは「まだ働いている最中でんがな。そないにせかさんといてや」という意味である。注意しておくがI/OのFF8$_H$番地は

OUTしたときはコマンドレジスタ
INしたときはステータスレジスタ

となっている。1番地2役なのだ。

このプログラムでは，ステータスの内容をチェックする意味で，いちいちプリントさせている。実行結果は，図1のようになるはずである。ちょっと，見てみる。

まず“]”は，「WNBSYが始まったよ」という意味である。最初のやつは，モーターオンから定常状態になるまで待っているわけだから，案の定A0$_H$=10100000$_B$で第7ビットが1である。いっしょに第5ビットも立っているが，これは「ヘッドがディスクに押しつけられている」ということを示している——はずだが，どっこい。実はX1ではこの信号は何の意味もないのであった。

ちょっと深入りするが，FDC↔FDD間にはHLDとHLTという信号がある。HLDはFDCからFDDへの「ヘッドをディスクに押しつけなさい」という信号である。HLT

リスト1

```
100 OUT &HFFC,&H81          :'MOTOR ON "1:"
110      GOSUB"WNBSY"
120 OUT &HFF8,&H2           :'RESTORE ( 第0トラックへシーク)
130      GOSUB"WNBSY"
140 OUT &HFFB,20            :'データレジスタ←目的トラック番号
150 OUT &HFF9,0             :'トラックレジスタ←現在のトラック番号
160 OUT &HFF8,&H1E          :'SEEK
170      GOSUB"WNBSY"
180 OUT &HFF8,&H5A          :'STEP IN
190      GOSUB"WNBSY"
200 OUT &HFFC,&H1
210 END
220 LABEL"WNBSY"
230 PRINT"]";SPACE$(3);
240 STAT=INP(&HFF8)         :'GET STATAS BYTE
250 PRINT RIGHT$("0"+HEX$(STAT),2);SPACE$(2);
260 IF STAT AND &H81 THEN 240
270 PRINT":":RETURN
```

リスト2

```
100 INPUT "STEP RATE";S:IF (S<0) OR (S>3) THEN 100
110 OUT &HFFC,&H81          :'MOTOR ON "1:"
120      GOSUB"WNBSY"
130 OUT &HFF8,&H0+S         :'RESTORE ( 第0トラックへシーク)
140      GOSUB"WNBSY"
150 OUT &HFFB,39            :'データレジスタ←目的トラック番号=39
160 OUT &HFF9,0             :'トラックレジスタ←現在のトラック番号
170 OUT &HFF8,&H1C+S        :'SEEK
180      GOSUB"WNBSY"
190 IF INKEY$(0)="" THEN 130
200 OUT &HFFC,&H1
210 END
220 LABEL"WNBSY"
230 PRINT"]";SPACE$(3);
240 STAT=INP(&HFF8)         :'GET STATAS BYTE
250 PRINT RIGHT$("0"+HEX$(STAT),2);SPACE$(2);
260 IF STAT AND &H81 THEN 240
270 PRINT":":RETURN
```

図1

```
]   A0  A0  A0  A0  A0  A0  A0  A0  A0  A0  A0  A0  A0  A0
A0  A0  A0  A0  A2  A0  A0  A0  A0  A0  A0  A0  A0  A0  A0
A0  A0  20  :
]   01  01  01  01  01  01  01  01  01  01  01  01  01  01
01  01  01  01  01  01  01  01  01  01  05  04  :
,   21  21  21  21  21  21  21  21  21  21  21  21  21  21
21  21  21  21  21  21  21  21  21  21  21  21  20  :
]   21  20  :
```

はFDDからFDCへの信号で「私ことFDDはディスクにヘッドを押しつけました。慎しんでご報告いたします」というものである。しかしX1では，HLDはオープン，HLTは5Vへプルアップされているのだ。すなわち，HLD信号は何にも使われておらず，HLTは常にHighのままである。これはどういうことかというと，5½インチ以下のサイズのフロッピーディスクドライブでは一般的に「ヘッドは常にディスクに押しつけられているから」なのだ。確認していないが，X1のFDDもそうなっているはずである。

よって第5ビットは無視すべきである。

さて，図1の2行目中ほどに「A2」というのが1カ所出てくる。これは第1ビットが立っているのである。表3-1から「INDEX」とわかる。すなわち，FDCはこの瞬間にインデックスホールを検出したのだ。あとはずーっとA0Hで，最後に第7ビットが0になり「NOT REAEYではない」。すなわち「用意できたよーん」となってWNBSYからリターンするのである。言い遅れたが，このようなステータスになるのは，直前（最後）にFDCに送られたコマンドがTYPE Iのコマンドだったからである。しかし，TYPEがなんであっても「NOT READY」と「BUSY」は同じことだから，悩む必要はない。

次にWNBSYに来るのは，RESTORE（リストア：トラック0へのシーク）のときである。01H=00000001Bと，BUSYが並んでいる。最後のほうは，05H，04Hとなっている。05Hは表3-1からわかるように「TRACK00ですAND BUSY」である。ちょっと待て！てめー，すでにTRACK0に来てるんなら，RESTORE動作は終わってんじゃねーか。それなのにBUSYとはなにごとだと一瞬逆上するが，次の04HでFDCは，BUSYを取り下げている。むーん。それにしても第5ビットが0だ――いったいどーなってんだ。ま，シカト（無視）を決め込むことにするのである。

次がシークコマンド後のWNBSYである。おっと，21H=00100001Bだ。第5ビットが立っている。いやいやシカトシカト。えーと，こっちは平凡にBUSYが0になって終わりである。

最後がステップ・インである。これはシークと同じであるが，移動距離が短いのであっという間に終わっている。

以上で図1の実感放送を終わる。

ではまず，シークコマンドについて少々補足しておく。シークコマンドでは，

データレジスタ ←目的トラック番号

トラックレジスタ←現在のトラック番号

とセットするわけだが，どーしてこんな面倒なことをするかというと，FDCが何台ものFDDを管理しているからなのだ。つまり，何台ものFDDを操作させているのだから，「現在のトラック番号」はX1の場合全部で4つある。本来，トラックレジスタは移動後の「トラック番号」を保持しているので，1台のドライブだけを動かしている場合は，目的トラック番号をデータレジスタにセットするだけでOKなのである。しかし現実はほとんどのシステムでFDDは2台以上あるから，DOSは4バイトのワークエリアを持っていて，いちいちトラック番号を保持，書き込みしてやらなければならない。

トラックレジスタがトラック番号を持っていることを確認するためには，

175 PRINT INP(&HFF9)

195 PRINT INP(&HFF9)

を入れてやれば良い。“20”，“21”と表示されるはずである。

次に大事な「フラグ」について説明する。リストアには，h, V, r_1，r_0の4つのフラグがあり，各動作を指定している。まずは表2-1を見ると，

h=ヘッドロードフラグ

とある。これはさっき言ったように無意味である。次に

V=トラック照合フラグ

である。これは実際にディスクを読んで，チェックするかどうかのフラグである。ディスクには「私は第△トラックの第×サイドの第○セクタです」という「ID」が書かれているのだ。これを読んで，ヘッドが本当に目的のトラックに達したかどうかのチェックを指定するフラグである。リスト1では02HだからV=0で，チェックは行ってない。これについては，あとで実験をする。

次にr_1，r_0であるが，これは2つで1組で，00B～11Bまでの4つの状態があり，これはステップレートを指定するものである。

リスト3

```
                                  .Z80
                                  .PHASE  0DF00H
                        ;
0FF8                    CR        EQU     0FF8H    ;COMMAND REG.
0FF8                    STR       EQU     0FF8H    ;STATUS REG.
0FF9                    TR        EQU     0FF9H    ;TRACK REG.
0FFA                    SCR       EQU     0FFAH    ;SECTOR REG.
0FFB                    DR        EQU     0FFBH    ;DATA REG.
0FFC                    MSDR      EQU     0FFCH    ;MOTOR,SIDE,DRIVE# SELECT
                        ;
DF00    D5              START:    PUSH    DE       ;SAVE DE FOR RETURN STATUS
DF01    1A                        LD      A,(DE)   ;GET COMMAND No.
DF02    13                        INC     DE
DF03    D5                        PUSH    DE
DF04    21 DF1A                   LD      HL,JTA   ;HL=JUMP TABLE AREA
DF07    16 00                     LD      D,0
DF09    87                        ADD     A,A
DF0A    5F                        LD      E,A      ;DE=A*2
DF0B    19                        ADD     HL,DE
DF0C    5E                        LD      E,(HL)
DF0D    23                        INC     HL
DF0E    56                        LD      D,(HL)   ;GET JUMP ADDRESS
DF0F    EB                        EX      DE,HL    ;HL=JUMP ADDRESS
DF10    D1                        POP     DE
DF11    F3                        DI
DF12    CD DF19                   CALL    PATCH    ;CALL (HL)
                        ;
DF15    FB                        EI
DF16    D1                        POP     DE       ;GET STRING ADDRESS
DF17    12                        LD      (DE),A   ;STORE LASR STATUS
DF18    C9                        RET              ;RETURN TO BASIC
                        ;
DF19    E9              PATCH:    JP      (HL)     ;CALL (HL)
                        ;
                        ;         Jump Table Area
                        ;
                        ;TYPE I
DF1A    DF32            JTA:      DW      RSTR     ;RESTORE
DF1C    DF3C                      DW      SEEK     ;SEEK
DF1E    DF55                      DW      STEP     ;STEP
DF20    DF55                      DW      STPIN    ;STEP IN
DF22    DF55                      DW      STPOT    ;STEP OUT
                        ;
                        ;TYPE II
DF24    DF5F                      DW      READD    ;READ DATA
DF26    DF84                      DW      WRITD    ;WRITE DATA
```

ステップレートは表2-1に示してある「時間」である。なぜこんなものがあるかというと，これは「FDDのヘッドを動かす速さ」に機種ごとで差があるからなのだ（一般的に高いFDDほど速い）。FDCはFDDに「ヘッドを動かしなさい」という信号を送るのだが，その信号は「方向（内側へか外側へか）」と「動け」だけなのだ。FDDは一般にタコだから，FDCのほうでタイミングをとってやらないといけない。そのタイミングを指定するのがr_1，r_0なのである。たとえばリスト1のように$r_1=1$，$r_0=0$を指定すると，FDCはFDDに対して20ms（0.02秒）ごとに「動け」と指令を出すことになる。さて，実はX1ではステップレートは別に20msでなくても良いのである。

3インチ＝3ms

turboの内蔵＝6ms

古いタイプ（レバーのないやつ）＝20ms

となっている。他にX1F model 20内蔵のやつとか，いろいろあるが，確認はしていないが，多分それらは6msだと思う。そこでリスト2が「ガリガリプログラム」である。指定したステップレートで，ひたすらトラック0↔39間を動くというものである。しつこくやったり，性能以上の速度で動かそうとすると故障の原因になるからほどほどにやること。性能を越えた場合は“32”とか“30”(第4ビット＝シーク・エラーが1）などが表示されるはずである。

あとに残っているのは，STEP等の

u＝トラックレジスタ更新フラグ

である。これは「ヘッドの移動に伴って，トラックレジスタの値を増減させるか，させないか」のフラグである。リスト1で180行の&H5Aを&H4Aにして，

195 PRINT INP(&HFF9)

を追加すると“20”と表示されるはずである。実際には，ヘッドは第21トラック上にあるのだが，トラックレジスタはそうなっていないのである。

なぜこんなフラグがあるかというと，フロッピーディスクの始祖，IBMフォーマットには「欠陥トラック」というものがあるためなのだ。たとえば，ディスクをフォーマットしているときにどうしても正常に書き込めないトラックを発見したとする。現在のパソコンではそんな場合には「エラーだよ，このディスクはダメ」とメッセージを

```
                        ;
                        ;TYPE III
DF28    DF5F                    DW      READI   ;READ ID
DF2A    DF5F                    DW      REDTR   ;READ TRACK
DF2C    DF84                    DW      WRITR   ;WRITE TRACK
                        ;
                        ;TYPE IV
DF2E    DFBC                    DW      FORI    ;FORCE INTERRUPT
                        ;
                        ;MOTOR,SIDE,DRIVE#
DF30    DFC6                    DW      MSD     ;MOTER,SIDE,DRIVE#
                        ;
DF32    1A              RSTR:   LD      A,(DE)  ;COMMAND
DF33    01 0FF8                 LD      BC,CR   ;BC=COMMAND REG.
DF36    ED 79                   OUT     (C),A
DF38    CD DFD2                 CALL    WNBSY   ;WAIT READY
DF3B    C9                      RET
                        ;
DF3C    EB              SEEK:   EX      DE,HL
DF3D    7E                      LD      A,(HL)  ;A=COMMAND
DF3E    23                      INC     HL
DF3F    56                      LD      D,(HL)  ;D=TRACK# TO SEEK
DF40    23                      INC     HL
DF41    5E                      LD      E,(HL)  ;E=CURREND TRACK#
DF42    01 0FF9                 LD      BC,TR
DF45    ED 59                   OUT     (C),E   ;SET CURREND TRACK#
DF47    01 0FFB                 LD      BC,DR
DF4A    ED 51                   OUT     (C),D   ;SET TRACK TO SEEK
DF4C    01 0FF8                 LD      BC,CR
DF4F    ED 79                   OUT     (C),A   ;SEND SEEK COMMAND
DF51    CD DFD2                 CALL    WNBSY   ;WAIT 'NOT BUSY'
DF54    C9                      RET
                        ;
DF55                    STEP:
DF55                    STPIN:
DF55    1A              STPOT:  LD      A,(DE)  ;GET COMMAND
DF56    01 0FF8                 LD      BC,CR
DF59    ED 79                   OUT     (C),A
DF5B    CD DFD2                 CALL    WNBSY
DF5E    C9                      RET
                        ;
DF5F                    READD:
DF5F                    READI:
DF5F    CD DFEA         REDTR:  CALL    SETSCT  ;A=COMMAND,SET SECTOR REG.
DF62    01 0FF8                 LD      BC,CR   ;CR=STR
DF65    D9                      EXX
DF66    01 0FFB                 LD      BC,DR   ;BC'=DATA REG.
DF69    2A DFFE                 LD      HL,(BUFAD)      ;HL'=BUFF ADD.
DF6C    D9                      EXX
DF6D    ED 79                   OUT     (C),A   ;SET COMMAND
DF6F    CD DFE4                 CALL    WAIT1   ;STRANGE ?
DF72    ED 78           RED1:   IN      A,(C)   ;GET STATUS
DF74    0F                      RRCA            ;CHECK BUSY
DF75    30 0B                   JR      NC,RED2 ;END READ
DF77    0F                      RRCA            ;CHECK DATA REQUEST
DF78    30 F8                   JR      NC,RED1 ;NO REQUEST
                        ;
DF7A    D9                      EXX
DF7B    ED 78                   IN      A,(C)   ;GET DATA
DF7D    77                      LD      (HL),A  ;STORE DATA
DF7E    23                      INC     HL      ;INC. BUFF. ADD.
DF7F    D9                      EXX
DF80    18 F0                   JR      RED1    ;AGAIN
                        ;
DF82    07              RED2:   RLCA            ;BACK STAT (RRCA <-> RLCA)
DF83    C9                      RET
                        ;
DF84                    WRITD:
DF84    CD DFEA         WRITR:  CALL    SETSCT  ;A=COMMAND,SET SECTOR REG.
DF87    01 0FF8                 LD      BC,CR   ;CR=STR
DF8A    D9                      EXX
DF8B    01 0FFB                 LD      BC,DR   ;BC'=DATA REG.
DF8E    2A DFFE                 LD      HL,(BUFAD)      ;HL'=BUFF ADD.
DF91    D9                      EXX
DF92    1E 00                   LD      E,00H   ;COUNTER
DF94    16 02                   LD      D,02H   ;MASK
DF96    ED 79                   OUT     (C),A   ;SET COMMAND
DF98    ED 78           FLOOP:  IN      A,(C)   ;FIRST LOOP
DF9A    A2                      AND     D
DF9B    C2 DFA4                 JP      NZ,FIRST        ;FIRST DATA REQ.
DF9E    1D                      DEC     E               ;DEC. COUNTER
DF9F    C2 DF98                 JP      NZ,FLOOP        ;AGAIN
DFA2    18 06                   JR      WRT1    ;QUIT FIRST LOOP
DFA4    D9              FIRST:  EXX
DFA5    7E                      LD      A,(HL)  ;GET FIRST DATA
DFA6    ED 79                   OUT     (C),A   ;WRITE 1 BYTE
DFA8    23                      INC     HL      ;INC. BUFF. ADD.
DFA9    D9                      EXX
DFAA    ED 78           WRT1:   IN      A,(C)   ;GET STATUS
DFAC    0F                      RRCA            ;CHECK BUSY
DFAD    30 0B                   JR      NC,WRT2 ;END WRITE
DFAF    0F                      RRCA            ;CHECK DATA REQUEST
DFB0    30 F8                   JR      NC,WRT1 ;NO REQUEST
                        ;
DFB2    D9                      EXX
DFB3    7E                      LD      A,(HL)  ;GET DATA
```

出して，別のディスクを使うように要求するのだが，IBMフォーマットでは，「交替用トラック」が2トラック分（本当はシリンダと表現すべきだが）用意されていて，多少の欠陥のあるディスクでも使えてしまうのである。ごていねいに，その欠陥のあるトラックを指定する方法も決まっている。この場合に，欠陥のあるトラックを飛ばしてフォーマットすると，実際のヘッドの位置と，ディスクに書かれている「ID」に違いが出てしまうことになる。そうなると，Vフラグでチェックしたときにシーク・エラーが出てしまうなどして都合がわるいので，uフラグを使って「つじつまを合わせる」のである。大昔は磁性体の質が低かったり，ディスク（メディア）も高かったりしたから，そんなことをしたのだろうが，今はほとんど意味がないと言えるフラグである。しかし，IBMフォーマットがいかに「金科玉条」であるかの良い証拠でもある。

というところで，少々遊んでみることにする。まずはフォーマットされてないまっさらのディスクを用意してほしい。それをドライブ1に入れていただきたい。その状態で，リスト1を実行すると，3番目のシークコマンドのステータスで，最後に30Hが出るはずである。すなわちシーク・エラーである。これは別にフォーマットされているディスクでも，ステップ・インで，u=0，Vフラグ=1（&H4E）とすると同じことが起きる。トラックレジスタの値は“20”なのに，Vフラグにしたがってチェックしてみると，第“21”トラックだった，というわけである。

TYPE Iコマンドで言い残したのは，STEPコマンドだけである。このコマンドを実行すると，ヘッドは内/外どっちに動くのかということであるが，「直前に動いたほう」に動くということである。すなわち，FDCは「最後に動いた方向」を保持しているというわけである。

TYPE IIの実習

TYPE Iでは機械語を使う必要がなかったが，TYPE IIではさすがにそうはいかない。そこで，リスト3とリスト4である。例によってリスト3は機械語プログラムのアセンブルリスト，リスト4はそれをBASICから使うためのプログラムである。BA

```
DFB4    ED 79                     OUT     (C),A     ;WRITE DATA
DFB6    23                        INC     HL        ;INC. BUFF ADD.
DFB7    D9                        EXX
DFB8    18 F0                     JR      WRT1      ;AGAIN
                          ;
DFBA    07                WRT2:   RLCA              ;BACK STAT (RRCA <-> RLCA)
DFBB    C9                        RET
                          ;
DFBC    1A                FORI:   LD      A,(DE)    ;GET COMMAND
DFBD    01 0FF8                   LD      BC,CR
DFC0    ED 79                     OUT     (C),A
DFC2    CD DFD2                   CALL    WNBSY
DFC5    C9                        RET
                          ;
DFC6    1A                MSD:    LD      A,(DE)    ;GET DATA
DFC7    01 0FFC                   LD      BC,MSDR
DFCA    ED 79                     OUT     (C),A     ;SET IT
DFCC    B7                        OR      A         ;MOTOR ON?
DFCD    F0                        RET     P         ;MOTOR OFF THEN RET
DFCE    CD DFD2                   CALL    WNBSY     ;WAIT READY
DFD1    C9                        RET
                          ;
DFD2    C5                WNBSY:  PUSH    BC        ;SAVE BC
DFD3    06 20                     LD      B,20H
DFD5    10 FE             WNBSY0: DJNZ    WNBSY0    ;WAIT
DFD7    01 0FF8           WNBSY1: LD      BC,STR    ;DASAI
DFDA    ED 78                     IN      A,(C)
DFDC    4F                        LD      C,A       ;SAVE STATUS
DFDD    E6 81                     AND     81H       ;CHECK
DFDF    20 F6                     JR      NZ,WNBSY1
DFE1    79                        LD      A,C       ;GET STATUS
DFE2    C1                        POP     BC        ;GET BC
DFE3    C9                        RET
                          ;
DFE4    3E 07             WAIT1:  LD      A,7       ;WAIT ROUTINE
DFE6    3D                WAIT2:  DEC     A
DFE7    20 FD                     JR      NZ,WAIT2
DFE9    C9                        RET
                          ;
DFEA    EB                SETSCT: EX      DE,HL
DFEB    7E                        LD      A,(HL)    ;A=COMMAND
DFEC    23                        INC     HL
DFED    56                        LD      D,(HL)    ;D=SECTOR#
DFEE    01 0FFA                   LD      BC,SCR    ;BC=SECTOR REG.
DFF1    ED 51                     OUT     (C),D     ;SET SECTOR#
DFF3    F5                        PUSH    AF        ;SAVE COMMAND
DFF4    CD DFD2                   CALL    WNBSY
DFF7    F1                        POP     AF        ;GET COMMAND
DFF8    C9                        RET
                          ;
DFF9    00 00 00 00               DB      0,0,0,0,0          ;DUMMY
DFFD    00
DFFE    E000              BUFAD:  DW      0E000H    ;BUFFER ADDRESS
                                  END
```

リスト4

```
100 CLEAR &HDF00
110 MEM$(&HDF00,16)=HEXCHR$("D5 1A 13 D5 21 1A DF 16 00 87 5F 19 5E 23 56 EB")
120 MEM$(&HDF10,16)=HEXCHR$("D1 F3 CD 19 DF FB D1 12 C9 E9 32 DF 3C DF 55 DF")
130 MEM$(&HDF20,16)=HEXCHR$("55 DF 55 DF 5F DF 84 DF 5F DF 5F DF 84 DF BC DF")
140 MEM$(&HDF30,16)=HEXCHR$("C6 DF 1A 01 F8 0F ED 79 CD D2 DF C9 EB 7E 23 56")
150 MEM$(&HDF40,16)=HEXCHR$("23 5E 01 F9 0F ED 59 01 FB 0F ED 51 01 F8 0F ED")
160 MEM$(&HDF50,16)=HEXCHR$("79 CD D2 DF C9 1A 01 F8 0F ED 79 CD D2 DF C9 CD")
170 MEM$(&HDF60,16)=HEXCHR$("EA DF 01 F8 0F D9 01 FB 0F 2A FE DF D9 ED 79 CD")
180 MEM$(&HDF70,16)=HEXCHR$("E4 DF ED 78 0F 30 0B 0F 30 F8 D9 ED 78 77 23 D9")
190 MEM$(&HDF80,16)=HEXCHR$("18 F0 07 C9 CD EA DF 01 F8 0F D9 01 FB 0F 2A FE")
200 MEM$(&HDF90,16)=HEXCHR$("DF D9 1E 00 16 02 ED 79 ED 78 A2 C2 A4 DF 1D C2")
210 MEM$(&HDFA0,16)=HEXCHR$("98 DF 18 06 D9 7E ED 79 23 D9 ED 78 0F 30 0B 0F")
220 MEM$(&HDFB0,16)=HEXCHR$("30 F8 D9 7E ED 79 23 D9 18 F0 07 C9 1A 01 F8 0F")
230 MEM$(&HDFC0,16)=HEXCHR$("ED 79 CD D2 DF C9 1A 01 FC 0F ED 79 B7 F0 CD D2")
240 MEM$(&HDFD0,16)=HEXCHR$("DF C9 C5 06 20 10 FE 01 F8 0F ED 78 4F E6 81 20")
250 MEM$(&HDFE0,16)=HEXCHR$("F6 79 C1 C9 3E 07 3D 20 FD C9 EB 7E 23 56 01 FA")
260 MEM$(&HDFF0,16)=HEXCHR$("0F ED 51 F5 CD D2 DF F1 C9 00 00 00 00 00 00 E0")
270 DEFUSR0=&HDF00
280  D$=USR0(CHR$(11,&H81))      :'MOTOR ON
290  D$=USR0(CHR$(0,&H2))        :'RESTORE
300  D$=USR0(CHR$(1,&H1E,2,0))   :'SEEK
310 'D$=USR0(CHR$(2,&H3A))       :'STEP
320 'D$=USR0(CHR$(3,&H5A))       :'STEP IN
330 'D$=USR0(CHR$(4,&H7A))       :'STEP OUT
340  D$=USR0(CHR$(5,&H80,1))     :'READ DATA
350 'D$=USR0(CHR$(6,&HA0,1))     :'WRITE DATA
360 'D$=USR0(CHR$(7,&HC0,1))     :'READ ADD.
370 'D$=USR0(CHR$(8,&HE0,1))     :'READ TRACK
380 'D$=USR0(CHR$(9,&HF0,1))     :'WRITE TRACK
390 'D$=USR0(CHR$(10,&HD0))      :'FORCE INT.
400  D$=USR0(CHR$(11,&H1))       :'MOTOR OFF
410 END
```

表５（リスト４の使い方）

コマンド	番号	第２パラメータ	第３パラメータ	第４パラメータ
リストア	0	コマンド	—	—
シーク	1	〃	目的シリンダNo.	現在のシリンダNo.
ステップ	2	〃	—	—
ステップ・イン	3	〃	—	—
ステップ・アウト	4	〃	—	—
リードデータ	5	〃	セクタNo.	—
ライトデータ	6	〃	セクタNo.	—
リードアドレス	7	〃	ダミー	—
リードトラック	8	〃	ダミー	—
ライトトラック	9	〃	ダミー	—
フォースインタラプト	10	〃	—	—
モーター，サイド，ドライブナンバー	11	データ	—	—

SICはCZ-8 FB01のバージョン1でなければならない。なぜかと言うと，場合によってはE000$_H$～F800$_H$ぐらいまでをワークエリアに使ってしまうからである。

まずリスト４の使い方からである。

基本的なパターンは，

番号(0～11)，コマンド，パラメータ…をCHR$で文字列に変換してUSR0（～)である。それは表５に詳しく示す。11番だけはFDCへのコマンドではなく，FFC$_H$番地へのデータであることに注意。

なお，このプログラムは手抜きのお手本みたいなもので，ちゃんとしたエラー処理をしていない。たとえばディスクを抜いてあるドライブのモーターをONにしようとすると，帰って来ないのである。そのときは心静かにリセットスイッチを押していただきたい。

では，おもむろにリスト３の解説を始める。

DF00$_H$～DF19$_H$は，０～11番のうちのどこへ飛び込むかを計算している。最初にDEレジスタをPUSHしているのは，のちのちステータスを返すためである。DF11$_H$に「DI」があって割り込みを禁止しているが，これはディスクリード／ライトのようにタイミングが大事なプログラムでは必要なことである。DF12$_H$ではDF19$_H$の「JP (HL)」とともに，「CALL (HL)」に相当することをやらせている。実際のルーチンは後回しにして，DF15$_H$～DF18$_H$を説明する。ディスクにアクセスして帰って来たときに，Aレジスタは最後に読み出したステータスレジスタの値を持っているようにしてある。その値をDEの指すアドレスにストアすると，このプログラムで呼び出したBASICで，

D$＝USR0(～)

となっていると，D$の先頭の文字として受け取ることができる。便利，便利。

では，実行ルーチンのほうであるが，たとえばDF32$_H$のRESTOREの実行ルーチンでは，「LD A, (DE)」によって，FDCに送るコマンドをAレジスタに拾い上げている。それをコマンドレジスタ（CR）にOUTして，WNBSYをコールする。よーするにこれは前述のBASICと同じである。SEEK, STEP等も同様。

次にDF5F$_H$からのREADD（リードデータ），READI（リードアドレス），REDTR（リードトラック）の３種混合ルーチンである。恐ろしいことにこれら3つのルーチンは完全に同じなのである。その結果，READI, REDTRでは，セクタ番号を指定する必要がないのに指定している。実害はないから，大丈夫。まず「CALL SET SCT」でAにコマンドを拾いあげ，なおかつセクタレジスタ（SCR）にセクタ番号をOUTしている。その先が少々複雑なのだが，BCレジスタにステータスレジスタ（STR＝CR）のアドレスをセットし，EXXで裏レジスタにして，BC'にデータレジスタ(DR), HL'にバッファアドレスをセットしている。ディスクのリードライトのためには，２つの違うI/Oアドレスに高速にアクセスしなければならないのでこうしたのだが，STRとDRの上位アドレスはともに0F$_H$だから，どうしても裏レジスタを使わなければならないわけではない。結局は趣味の問題である。そのあとはコマンドをCRにOUTして，少々待つ。なぜかよく知らないが，とにかく待つ。それからはレジスタを表にしたり裏にしたりしつつ，データを受け取るのである。DF82$_H$番地のRLCAは右にローテートしたステータスを左にローテートして元に戻すものである。

次にDF84$_H$からの書き込むほうであるが，DF98$_H$～DFA9$_H$間は，１バイトだけ書くルーチンである。なんでこんなことをしているかというと，ライトトラックのためなのだ。なぜかこうしないとライトトラックが成功しないのである。理由はよくわからないが動けば良いのだ。他は基本的にリードルーチンと同じである。

さて，リスト４を実行すると，ドライブ１に対してリストア動作したのち（290行），第２トラックへシークし（300行），第１セクタをリードし（340行），モーターをOFFする。データを読み込む番地は，DFFE$_H$番地からの２バイトに格納されているアドレスで，リスト４ではE000$_H$番地になっている（260行の右端)。D$の先頭バイトには，最後に読み込んだステータスレジスタの値が入っているから，

PRINT ASC(D$)

で，エラーチェックもできるようになっている。そこでさっそくリードデータのかわりに，370行のリードトラックを実行してみていただきたい。「D$＝～」のかわりに，「D1$＝～」とでもしておくべきだろう。実行後，PRINT ASC(D1$)とすると，"4"が表示されるはずである。ステータス表を見ると，なんと「LOST DATA」である。これは結局どーゆーことかと言うと「さっさと読み出さなかったから，データの取りこぼしがあるよ」ということである。つまりはエラーである。

話せばわかる！ X1では，どーしてもこうなるのである。これはしかたないのである。私のプログラムのせいではないのである。しかし，ここにも一筋の光明があるのだ。X1turboの内蔵ディスクならば，「LOST DATA」にはなるが，十分にデータは読み出せているのである（来月やるのだ)。他のディスクについては，３インチと800F等の旧タイプでは，データがビットずれを起こしてしまって使いものにならないことを確認済みである。CZ-501FやX1Fの内蔵ドライブは，申しわけないが確認していない。

しかし，リードトラックよりも大事なことがあるのだ（と逃げる)。まずはTYPEIIコマンドのフラグについて説明しておく。

・m＝マルチレコードフラグ

これは，複数のレコードをいっぺんに読み書きするときに使うフラグである。あとで実験する。

・S＝サイドフラグ

これはCフラグといっしょに使うもので，読み出したディスクのサイドが０であるべ

きか，それとも１であるべきかを指定する。

・E＝ディレイフラグ

これは，ヘッドがディスクに押しつけられたかどうかの信号のサンプリングタイミングを指定するもの。X1では意味がない。

・C＝サイド番号比較フラグ

これはVフラグがトラック（シリンダ）番号を比較するかどうかのフラグであるのに対して，サイド番号を比較するかどうかのフラグである。しつこいようだが，Sフラグといっしょに使う。

・a0＝アドレスマークフラグ

これはライトデータのときにだけ指定できるものである。これはサンプルを示したほうが理解しやすいだろう。

では，TYPEIIの実習である。図２である。

まずはmフラグをいじってみる。リスト４の280行から図2-1のように書き換えていただきたい。まずは280行でモータオン，ドライブは“1：”でサイドは０番である。リストアして，m＝1で第１セクタから読み始めている。325行で，それぞれのステータスレジスタを表示している。その値は，

リストア命令→4(TRACK 00＝1)

リードデータ命令→16(RECORD NOT FOUND)

となっている。モニタに飛んでダンプしてみると，しっかり16セクタ分が読み込まれているはずである。そこで，BASICに返り，

PRINT INP(&HFFA)

として，セクタレジスタを表示してみると“17”と出るはずである。すなわちFDCはmフラグ＝1に従って，第１セクタから第16セクタまでを読み，あげくの果てに第17セクタまで読もうとしたが「無い袖は振れない」の法則に従って，当然のごとく失敗して，「RECORD NOT FOUND」を起こしたというわけである。このmフラグを使うと，連続セクタを読むという点ではなかなかに高速なのだが，おしむらくは「いつ終わるのかを指定できない」という欠点があるのだ。それでFDCは「第17セクタはどこだっ!?」と，少しの間捜し回るので結局は時間を食ってしまう。よって正しく使うには，必要なだけを読んだかどうかをチェックしてやって，TYPEⅣのフォースインタラプトを使って，コマンドを打ち切ってやらなければならない。なかなかに世話のやけるmフラグであった。合掌。

次にS，Cフラグである。図2-2は，平和な例で，280行でサイド０を指定しているから，S＝0（サイドは０ですか？），C＝1（チェックしてください）によって，ステータスは０という値になる。この逆が図2-3で，280行でサイド0を指定しているのに，S＝1（サイドは１ですか？），C＝1（チェックしてください）としているので，しっかり16＝RECORD NOT FOUNDが出ている。この場合はデータの読み込みは行われない。

a0フラグの実習が図2-4である。a0フラグ＝1で，ライトデータを実行すると，そのセクタにデリーテッドアドレスマークが書かれてしまうのだ。具体的にどこかというのは来月やるのである。これを検出するには，そのセクタを読むだけで良い。図2-4のように，ステータスレジスタが32となり，第５ビット（RECORD TYPE）が１になる。この場合にデータはちゃんと読まれるから，特にチェックしないかぎり同じことである。では，なぜこのようになっているかというと，例によって「IBMに聞いてくれ」なのであった。『X１マシン語活用百科』(産業報知センター）には，これを「プロテクトに使える」と書いてあって，なるほどと思うが，実社会におけるプロテクト事情は，さらに生き馬の目をくり抜くようである。

以上で今月はひとまず終わりである。来月は，ありとあらゆる変質的なFD操作術へと食指を伸ばす予定である。最後になったが今月（と来月）の本文に対して，その筋の牛嶋氏から貴重な助言をいただいた。とりあえずここに感謝のポーズ。ではまた来月。

図2-1

```
m FLAG

280  D$=USR0(CHR$(11,&H81))          :'MOTOR ON
290  D0$=USR0(CHR$(0,&H2))           :'RESTORE
300  D5$=USR0(CHR$(5,&H90,1))        :'READ DATA
310 'D6$=USR0(CHR$(6,&HA0,1))        :'WRITE DATA
320  D$=USR0(CHR$(11,&H1))           :'MOTOR OFF
325 PRINTASC(D0$),ASC(D5$),ASC(D6$)
330 END

 4          16          0
```

図2-2

```
S,C FLAG...S=0,C=1

280  D$=USR0(CHR$(11,&H81))          :'MOTOR ON
290  D0$=USR0(CHR$(0,&H2))           :'RESTORE
300  D5$=USR0(CHR$(5,&H82,1))        :'READ DATA
310 'D6$=USR0(CHR$(6,&HA0,1))        :'WRITE DATA
320  D$=USR0(CHR$(11,&H1))           :'MOTOR OFF
325 PRINTASC(D0$),ASC(D5$),ASC(D6$)
330 END

 4          0           0
```

図2-3

```
S,C FLAG...S=1,C=1

280  D$=USR0(CHR$(11,&H81))          :'MOTOR ON
290  D0$=USR0(CHR$(0,&H2))           :'RESTORE
300  D5$=USR0(CHR$(5,&H8A,1))        :'READ DATA
310 'D6$=USR0(CHR$(6,&HA0,1))        :'WRITE DATA
320  D$=USR0(CHR$(11,&H1))           :'MOTOR OFF
325 PRINTASC(D0$),ASC(D5$),ASC(D6$)
330 END

 4          16          0
```

図2-4

```
a0 FLAG

280  D$=USR0(CHR$(11,&H81))          :'MOTOR ON
290  D0$=USR0(CHR$(0,&H2))           :'RESTORE
300  D6$=USR0(CHR$(6,&HA1,1))        :'WRITE DATA
310  D5$=USR0(CHR$(5,&H80,1))        :'READ DATA
320  D$=USR0(CHR$(11,&H1))           :'MOTOR OFF
325 PRINTASC(D0$),ASC(D5$),ASC(D6$)
330 END

 4          32          0
```

その筋質問箱

我解答者祝一平。今月唯一行。特急解答！

Q 挑戦状 最近S-OSのアプリケーションが増えてたいへんうれしいのですが，どうもあの長ったらしいダンプリストを入力するのがめんどくさく感じます。そこで僕は考えました。あの，縦横チェックサムを入力するだけで，自動的にダンプリストを出力することはできないだろうかと。もし，できるのでしたら，少々スピードが遅くても，ぜひ作ってください。この挑戦は，あなたをその筋の人と見込んでの挑戦です。　三重県　楠見武士

A ふむふむ。なかなかにその筋の発想であるが，もう少し修業が足りないよーである。Oh！MZのチェックサムは，縦16バイト，横8バイト＝128バイトに，24個のサムをつけたものである。ネライは，この24バイトを打ち込むだけで，128バイトを「作る」ということである。しかし，残念ながら「世の中はそんなに甘くないの法則」により，その手のことはできないのである。わかりやすく言うなら，たとえば32Kバイト＝32768バイトのプログラムを考えてみる。128バイトを24バイトで「作り出せる」なら，24/128＝3/16の手間ですむということである。よって，32Kバイトのプログラムは，32768×3/16＝6144バイトを打ち込むだけですんでしまう。しかしそーなると，さらにその6144バイトは6144×3/16＝1152バイトを打ち込むだけで「作り出せる」じゃないか。さらにはどどーっと，

1152→216＝128＋88→24＋88＝112

となり，結局は112バイトを打ち込むだけで，大枚32Kバイトのプログラムが「できて」しまうのであった。どーだ，いかにもあり得ないよーな気がしてきたであろう。

以上に述べたことをムズカシく言うと，「情報理論」という学問になるのである。縁起ものだから，ちょいとさわりだけ書くと，「情報（データ）にはエントロピーつうもんがあって，これはランダムさ（規則性の逆）を表す。それは計算できる」，「情報（データ）に規則性があればあるほど圧縮することができる」などなどとなっている。ざっと計算してみたところ，機械語プログラムは，せいぜい10％ぐらいしか圧縮できないようである。これは1回だけの圧縮で，90％→81％→…とはいかない。それが人生である。次の方どーぞ。

Q 祝先生こんにちは。僕はビリー・ジョエルが好きです。僕たちその筋一歩手前の人間たちの重要問題である“コピー”とか“だびんぐ”がよくこのコーナーで取り上げられていますが，メディアを純粋にアナログ信号として取り出せば，いかなる“ぷろてくと”も3秒で死ぬのではないのでしょうか？　リクエスト曲はオンリーヒューマンをお願いします。祝先生さようなら。　愛知県　中野智之

A 中川智哉氏が横で「グラスハウスは良いが，オネスティはイマイチだ」などとうるさいので，リクエストはしないよーに。

…
…

おっと，解答を忘れるところであった。えーと，アナログ信号としてコピーすれば良いのではないかということだが，そーはいかないのだ。なぜならアナログ信号というものは「劣化」するからなのだ。すなわちテープに記録されている信号は，⎍ なのだが，これをアナログ的にコピーしようとすると ⌒ になってしまうのである（正確に言うともう少し違うけど，信号が“丸く”なってしまうという基本は同じ）。そこでもうひとつひねると，「波形整形」というテクニックが出てくる。すなわち，⌒ になってしまった信号を，電気回路を使って⎍ に戻すのである。しかし，その方法とて完全無欠とはいかないよーで，最近のプロテクト技術はほとんど魔術に等しくなっている気配である。ま，とにかく近年ではアナログコピーなるワザは，ヘッドロックと同程度の効果しかないと見て間違いない。次の方どーぞ。

Q 一平さん，助けてください。僕はこの間までX1Cでせっせとミュージック・データを作り続けてきました。先日，めでたく愛機をX1 turboに買い換えたので，データをディスクに転送したのですが，次のような恐ろしい事態が発生してしまいました。従来のBASIC（CZ-8CB01）では，たとえば

```
10 SOUND 11,100:SOUND 12,100:
SUOND 13,0
20 PLAY "V16C9:V12E7G"
```

とやると，チャンネルAのエンベロープがちょん切れてしまいますが，

```
20' PLAY "V16C9:V12E7R"
```

ではちょん切れませんでした。

ところがturbo BASICでは後者もちょん切れてしまうのです。おかげで歯切れのよいベースラインをつけた曲などでは，エンベロープをかけた主旋律のリズムがムチャクチャになってしまいます。

なんで，従来のBASICとturbo BASICでPLAY文の反応が違うのですか？　turbo BASICで，エンベロープがちょん切れないようにするにはどうしたらいいですか？

なお，CZ-8FB01ではうまくいくのですが，フリーエリアが足りなくて，どうしてもturbo BASICで走らせたいのですが……。　北海道　大堀尚己

A おおっ！まさしくそのとおりの症状である。質問では「エンベロープがちょん切れる」と表現しているが，正確に言うと，8CB01では「20」だとチャンネルAのエンベロープが

——（2山）

となり，「20'」では

——（1山）

となるが，turbo BASICでは，どちらも2山になってしまうということなのである。BASICを解析したわけではないが，原因は，turbo BASICでは，「20'」のとき「R」のときも，PSGの13番レジスタ（エンベロープパターン指定）にデータを再設定するためらしい。'85年11月号にも書いたように，13番レジスタに書き込まれると，エンベロープが再発して，2山目がぴょこんと出てしまうのだ。しかし，BASICのマニュアルには，「V0～15」となっているので，V16はたまたまエンベロープモードになるというだけの，いわば隠し命令であるから，文句は言えないよーである。

さて対策であるが，私が思いついたのは残念ながら「“R”を書かない」ということだけである。書かなければチャンネルAをすべて演奏し終えるまでは，他のチャンネルは“R”を実行しているのと同じことになるので，それを使って切り抜けるのである。PLAY文を分割するなどであれこれいじれば，多分どうにかなると思う。しかし，私が勧めるのは，ミュージックエディタ（できれば楽譜も表示できるやつ）を，機械語で自作することである。——と誘惑する私であった。♬こっちのその筋はあーまいぞ。できたら投稿していただきたい。

というところで，今月はこれまでである。どこからでも，かかってきなさい！

THE SOFTOUCH SPECIAL ノミネート決定 1985

GAME OF THE YEAR

いつだって面白いゲームが必要だ。Oh！MZは独自の立場からGAME OF THE YEARを選出することにした。今月は，作品賞ほか各賞のノミネート作品を発表，読者諸君の審査を待つ。そして来たる3月号において記念すべき初代GAME OF THE YEARが誕生するのだ。さあ，君もゲーム史上に残るこの一大イベントに参加しよう。

Oh！MZはついに やってしまうんだゾと。

GAME OF THE YEARはいうまでもなくその年を代表する優れたゲームソフトを選ぶもので，パソコンゲーム大賞ともいうべきものだ。それは，もっとも面白いゲームであり，もっともパソコンゲームの発展に貢献したソフトである。このオシャレなソフトを選ぶため，Oh！MZではお馴染みの清水和人氏を始めとするGAME OF THE YEAR決定実行委員会を結成し，公正かつユニークな選出方法を設定した。まず，対象となる作品は，この1年間に話題になったすべてのパソコンゲームソフトとし，発売時期はあえて問題にしない。もちろん，実際に選出するのはOh！MZを読んでいるゲームファンに限られるため，ピンボールしか知らない98オジサンの票が集まる心配もない。そして，公正といえば，アカデミー賞のように配給映画会社の力関係がどうとか，レコード大賞のように所属事務所がどうたらこうたらといった問題が存在しない。スクウェアのパーティではローストビーフが美味しかったとか，アスキー宣伝部のK嬢は声がかわいいとかいった政治的背景もない。あるのは読者1人ひとりの正義の目であるといえるだろう。

それでは，GAME OF THE YEAR各賞の内容と具体的な選出方法を見て，ぜひとも君の情熱の1票を投票しよう。はたして時代が求めているのはどのゲームか？

作品賞
ウイングマン
ヴォルガード
EXOAII ウォーロイド
F2グランプリ
キングフラッピー
サラダの国のトマト姫
サンダーフォース
ザ・キャッスル
ザナドゥ
ザ・ブラックオニキス
ジャン狂
ゼビウス
タイムトンネル
大脱走
ちゃっくんぽっぷ
超時空要塞マクロス カウントダウン
テグザー
デゼニランド
デゼニワールド
デーモンクリスタル
TOKYOナンパストリート
ドラゴンスレイヤー
ドルアーガの塔
ハイドライド
ハイパーオリンピック'84
ファンタジアン
フラッピー
プラズマライン
ボコスカウォーズ
ミステリーハウス WORRY
野球狂
ラグランジュL2
ロードランナー

オリジナル シナリオ賞
サザンクロス
TOKYOナンパストリート
タイムトンネル
ラグランジュL2
暗黒城

特殊効果賞(サウンド＆グラフィック)
テグザー
ラグランジュL2
ハイドライド
ザナドゥ
プラズマライン

テーマ音楽賞
テグザー
タイムトンネル
ボコスカウォーズ
新竹取物語
ギャラガ

美術デザイン(キャラクター，背景)
妖怪探偵ちまちま
テグザー
ハイドライド
ファンタジアン
レリクス

主演キャラクター賞
フラッピー
テグザー
ちゃっくん(ちゃっくんぽっぷ)
ラファエル王子(ザ・キャッスル)
妖怪探偵ちまちま

助演キャラクター賞
エビーラ(フラッピー)
MELEON(ラプテック)
ミクちゃん(ウイングマン)
アンドアジュネシス(ゼビウス)
クラーケン(ザ・ブラックオニキス)

Oh！MZ賞
ちゃっくんぽっぷ
ドルアーガの塔
F2グランプリ
タイムトンネル
プラズマライン

Oh！CZ賞
ボコスカウォーズ
ゼビウス
ザナドゥ
ハイドライド
ハイパーオリンピック'84

アミューズメント・アクション部門賞
ヴォルガード
ロードランナー
テグザー
ザ・キャッスル
キングフラッピー

スポーツゲーム部門賞
ビクトリアスナイン
野球狂
ホールインワン
F2グランプリ
ハイパーオリンピック'84[1]

SF＆ファンタジー部門賞
ハイドライド
ラグランジュL2
ザナドゥ
ザ・ブラックオニキス
ファンタジアン

思考型ゲーム部門賞
棋太平
森田和郎のオセロ
フラッピー
ロードランナー
モール・モール

特別企画賞
ブラスティー
新竹取物語
ウイングマン
超時空要塞マクロス カウントダウン
サンダーストーム

移植ビデオゲーム賞
ゼビウス
ドルアーガの塔
ハイパーオリンピック'84
ギャラガ
マッピー

移植外国ゲーム賞
ウィザードリィ
ロードランナー
インポッシブルミッション
アズテックX

年間ゲームソフトBEST30

1. ファンタジアン
2. ハイドライド
3. ザ・ブラックオニキス
4. テグザー
5. ロードランナー
6. 野球狂
7. 超時空要塞マクロス カウントダウン
8. ゼビウス
9. F 2グランプリ
10. ザナドゥ
11. デーモンクリスタル
12. プラズマライン
13. タイムトンネル
14. ザ・キャッスル
15. ちゃっくんぽっぷ
16. フラッピー
17. デゼニワールド
18. 大脱走
19. サンダーフォース
20. EXOAII ウォーロイド
21. ハイパーオリンピック'84
22. ドラゴンスレイヤー
23. ミステリーハウス WORRY
24. ラグランジュL 2
25. ウイングマン
26. ヴォルガード
27. サラダの国のトマト姫
28. ジャン狂
29. ドルアーガの塔
30. デゼニランド

審査員ノミネートBEST10

1. ザ・ブラックオニキス
2. ハイドライド
3. テグザー
4. ファンタジアン
5. ザナドゥ
6. ゼビウス
7. ボコスカウォーズ
8. ザ・キャッスル
9. TOKYOナンパストリート
10. キングフラッピー

年間ゲームソフトBEST30決まる

年間ゲームソフトBEST30は，愛読者カードの，「推薦するパッケージソフト」の集計によるもので，1月号から12月号までの愛読者カードから各月 500通ずつ任意に抽出して集計を行った。結果はご覧のとおりである。第1位は，6月号ごろから異常な勢いで票を伸ばしたファンタジアンだ。また，予想どおりの強さを見せたハイドライドや，1年を通して安定した人気を保ったザ・ブラックオニキスなどが2位，3位につけている。いわゆる赤丸上昇中なのがテグザーとザナドゥである。テグザーは9月号，ザナドゥは11月号あたりからの追い上げでここまで来たのだ。

そのほか，スタンダードナンバーともいえるロードランナーやJOY JOY PACK にも採用された野球狂，そしてレーシングゲームの雄F 2グランプリなどもBEST10内に入っている。

ノミネートの方法

GAME OF THE YEAR 最大の関心はなんといっても「作品賞はどのソフトが獲得するか」だろう。この作品賞選出のベースになっているのは，毎月愛読者カードで寄せられる「推薦するパッケージソフト」である。まず，これをもとに集計した「年間ゲームソフトBEST30」とGAME OF THE YEAR 決定実行委員会による「審査員ノミネートBEST10」を発表することにした。

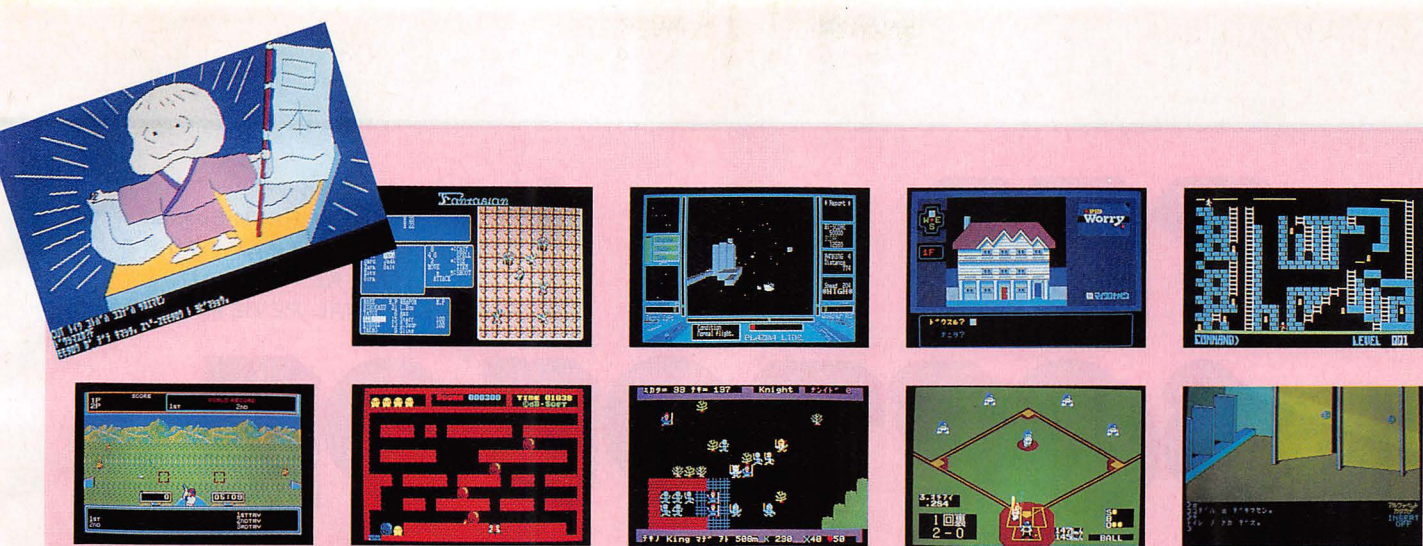
さて，実際のノミネートに関しては，読者の選んだBEST30に加えて，審査員ノミネートBEST10が加算され，その結果33本のゲームソフトが作品賞にノミネートされたわけである。

さあ，この33作品の中からGAME OF THE YEAR を選ぶのは読者自身にゆだねられる。ぜひとも，これはと思う作品を選んで投票してほしい。

また，その他の各賞に対してはGAME OF THE YEAR 決定実行委員会によるノミネートがなされた。ただし，ここにノミネートされなかったソフトへの乱入投票も認めることにする。

決戦投票の方法

ハガキに投票したい賞と作品名（何部門でもよい）を記入して下記の宛先へ送ってほしい。年賀ハガキや暑中見舞の余ったやつでかまわないが，愛読者カードは集計されないので注意してほしい。

〒102 東京都千代田区四番町2－1
㈱日本ソフトバンク
Oh！MZゲーム係　　1月15日消印有効

GAME OF THE YEAR,果たして作品賞は!?

■どんなかたちにせよ,No. 1 のゲームを決めるというのはワクワクしますね。もちろん読者投票で人気ソフトを選ぶというのはめずらしいことじゃないけど，ノミネート方式を採用したのはこれが初めてじゃないかな。できるだけ多くのソフトにチャンスをということで33本のソフトがノミネートされたわけだけど，ぼくら審査員のノミネートがそれほど影響するとは思えないし，読者が選んだBEST 30から予想すると上位 4 作品ぐらいの激戦になるんじゃないかと思ったりもする。あれほどの盛り上がりを見せたファンタジアンがそう簡単に 1 位の座を明け渡すとは思えない。そうかといってMZ-2000／2200／2500版の登場でハイドライドの逆転も考えられる。ザ・ブラックオニキスはザ・ファイヤークリスタルの登場がどこまで再燃に影響するかだろう。一方，RPGだらけの今年のソフトの中でテグザーの活躍は実に気持ちがよかったと思う。追い上げが面白くなりそうだ。さて，ぼく個人としてはザナドゥをBEST 3 にしているんだけど，時期的にみると難しいところかもしれない。むしろ来年にチャンスがあるというほうがいいかもしれないね。それとウィザードリィがノミネートされなかったのは不思議といえば不思議だ。なんとなくみんなも，いまさらウィザードリィを推薦してもどうか…って気持ちがあるのかも。これも来年には真価が問われることになるだろう。もっとも日本のゲームもいつまでもウィザードリィを目標にしていては困るけどね。

(斎藤　晋)

■ 1 年というのは短かいもんだ，もしくは，ゲームの流れは速いもんだ的発想の私でいくと，このゲームあのゲーム，それはそれなりにうわあ懐しいなんて考える。僕にとってそういう想い出のゲームv.s.今燃え上がるゲームを同じ土俵の上で冷静かつ過激に順位づけをするというのは簡単にいうと極度に刺激的だ。だから，いったいどれが 1 位になるかなんてこたあわからんが，こりゃ入らないなんてのはありたさん。ねらいわいいのになんかいまいちってやつあだめだね。少なくとも，たくさん売れたってより，なんかこうもっとうおーって感じで皆が殺到したのがきっと 1 位だよ。でもこれだけのゲームの中で 1 位になるっていやあすげえ。そんかわり「うっそー，あたし信じらんなあい」って人もいるってとこかな。オレはここんとこRPGにハマってるから，その辺を上位に押してみたジャン。ほかの人と違うのはナンパとリザードだろう。オラなんてったってこういうのが好きだべ。プレイしてるとなんかこう体さカーッと燃えてきてよう，SPELLとBITEとATTACKさいっぺんにくらったときみてえにしびれちまったあ。だけどちきしょう，なんであのゲームがあんなに低くて，よりによってこんなのが上のほうにいるんだろう。だんだんハラがたってきたぞ。よし，僕も年賀状の残りで投票しようっと。それから日本語は正しく使いましょう。じゃんじゃん。

(清水和人)

■僕はB型で気の長いほうではない。そういうわけだから，選ぶゲームも読者の皆さんとは大違いになるはず。そうです。他人のことなど考えられずに大切な花畑を平気で踏み荒らして，ひんしゅくを買うＢ型，ですからテグザーなんかは大嫌い。でも，みんなはテグザーが好きだ。う～む。ギャップに苦しんでしまう。どちらかというと，ロールプレイよりもアドベンチャーが好きだな。今はまだ，言葉探しの状態であるわけだけども，ストーリーを楽しむことができるアドベンチャーは僕にとってシューティングゲームの次に好きなジャンルです。そういうことでは，今年はロールプレイの嵐が吹きまくって僕は悲しかった。ただ，ハイドライドは唯一，暗いと言われてきたロールプレイに新風を巻き起こしたゲームだと言えるだろう。私は喜んだです。夢のかたまりであるパソコンには，夢のあるゲームがなくてはいけないのダ。ちなみに，順位の予想はＢ型の僕には聞かないよーに。僕は，ギャラガやマッピーが大好きなのダ。

(斎藤　亮)

■いきなりGAME OF THE YEARなんつうもんがおっぱじまってしまった。聞くところによると良いゲームを選び出してひょーしょーしてしまうというわけで，あんまり良くわかんないけど，なにかの間違いで日本のパソコンゲーム界の進歩に貢献するかもしれないので，とりあえず私も審査員になったしだいである。

私ははっきし言って，賞というものをあんまし「良いもの」とは思っていない人なのである。どーしてかっつーと「賞」というと日本のTV界が乱発している「輝け！　あーたらこーたらへまらか大賞！」とか，史上最強のパロディともいえる「日本アカデミー賞」などを思い浮かべてしまうのである。HONDAが常に最高のものに挑戦しているのとは大違いで，日本ではとりあえず「手の届く賞を作ってしまうもんねー」なのである。

だから何をいーたいんでーというと，私は良い審査員にはなれっこないということなのである。つまり悪徳レフェリーになってしまうのである。どーだすごいだろうと言いつつ私は，ときにはゆっくりとあるいは素早くカウントを取るのであった。

なにはともあれ，大事なのは賞ではなく,「本来なら賞に値するもの」なのである。だから私としては，いきなり「無冠の帝王賞」とかいいだしてツッパル予定である。

(祝　一平)

■GAME OF THE YEARのノミネートといっても個人の好みがあるから，これが絶対いい！　と言い切ることはできない。私の場合，たまたまBEEPというゲーム誌の仕事をしていてほとんどのゲームが見られるからということもあって，今回の審査員ノミネートに参加させてもらった。と，偉そうに言ったが，はっきり言ってやっぱり好みで選んでしまうのだ。私はRPGマニアなので，最近のよくできたRPGを上位に選んでみた。いずれも読者の支持の高いものになってしまい，おまえもミーハーだなと思われては審査員としての立場がない。そこで，あえて私はファンタジアンが嫌いなのだと言っておこう（立場はなお悪くなりそうだが…)。ファンタジアンは読者が推薦するソフトのナンバーワンだが，なぜ嫌いかというとオリジナリティがぜんぜん感じられないからなのだ。すでにウルティマIIIとウィザードリィが日本のパソコンでも動いているので，わかる人にはわかるはず。逆にアクションゲームのテグザーは素晴しい。パソコンでは沈みがちだったアクションゲーム界に新風を呼び起こした。本当なら 1 位にしたいぐらいだが，どうもテグザーにはSRのイメージがあるのでOh！MZで 1 位になってほしくない気もしているのだ。さあ君たちも独自の目と感覚でこれがGAME OF THE YEARという作品を選出してほしい。

(早川　浩)

THE SOFTOUCH

Part.1 ソフトでワイワイ

今月は新作SOFTの紹介のほかに，MZ-1500用JOY JOY PACKスペシャルの乱入もある。冬休みは思いっきりゲームに狂ってみるのもよいだろう。というわけで，今年もよいソフトがたくさん発売されますよーに。

MZ-2500のウィザードリィは発売されたか!?

ウィザードリィの日本語バージョンが発売された。肝心な移植の出来映えだが，APPLE版以来のウィザードリィファンの間でもかなり好評なようだ。ゲーム進行やプレイヤーそれぞれの冒険に影響するシナリオやマップは原作に忠実で，そして画面表示にはウィザードリィIIIと同じウインドウ型式にバージョンアップされている。しかも，ハイレゾグラフィックによるキャラクターデザインがなされ，視覚的にもグンと楽しめるようになっている。移植で苦労したのは日本語への翻訳といわれるが，この日本語版ウィザードリィでは，英語，ひらがな，カタカナの表示が選択できるのはうれしい。

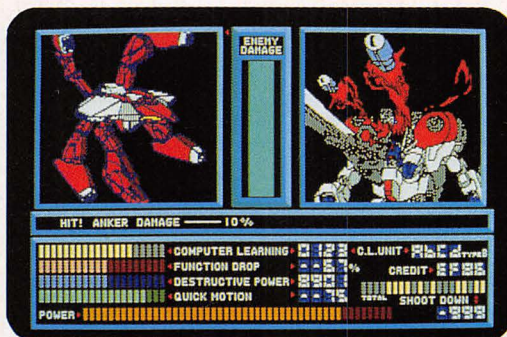
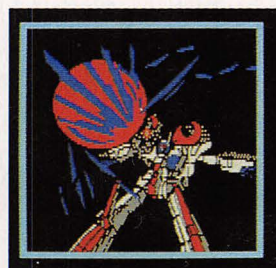
ウィザードリィは，X1/X1turbo版がアスキーから発売されている。しかし，このウィザードリィがSuper MZすなわちMZ-2500でも出るとしたらどうだろう。実は，Oh! MZにはウィザードリィのMZ-2500版がフォア・チューンから発売されるという情報が入り，その動向を見守ってきた。ところが，公式発表のないままとんでもないことが起こった。11月24日のパソコンサンデー(テレビ東京)におけるウィザードリィ紹介の際，なんとウィザードリィが動いているMZ-2500の姿が放映されたのだ。しかも，PC-9801バージョンと同じく，キャラクターネームが漢字表示になっていたのである。発売時期や価格については触れられなかったが，もしかしてこの正月はウィザードリィにハマってしまったMZ-2500ユーザーが続出するのではないだろうか。

X1/X1turbo用	5D：9,800円
アスキー	☎03(486)7111
MZ-2500用	3.5D：9,800円
フォア・チューン	☎03(667)2707

本格アニメ処理のRPGに期待　ブラスティー

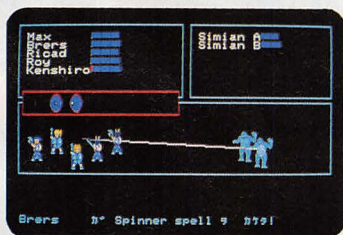
ウィル，テグザーと連続ヒットを続けるスクウェアが持ち前のアニメ処理技術をフルに生かした過激なゲームを発表した。この新作ブラスティーでは1秒間に6コマの絵を表示するアニメ処理が全編にほどこされている。そして，数百枚に及ぶアニメ原画を担当したのが，機動戦士ガンダムやダーティペアで知られる日本サンライズだ。2枚のディスクにテーマ曲を収録したソノシート，そしてその筋のマニアを狂喜させる豪華設定資料集やステッカーまで付いている。アニメファンの期待を担って12月末発売予定。

X1turbo用	5D：7,800円
スクウェア	☎044(63)6201

©日本サンライズ

ブラスティー

新作SOFTWARE

ザ・ファイヤークリスタル

ブラックオニキスの続編，ファイヤークリスタルのX1版が遂に発売された。今回は，テンプル編で魔法も使えるようになり，より正確な判断力と計画性が要求される。前作をも圧倒するパワーで面白さは保証付き。なお，ディスク版は近日発売の予定。

X1/X1turbo用　T：4,800円
BPS　☎045(421)7421

ザ・ブラックオニキス

日本のRPG史上に輝くザ・ブラックオニキスのMZ-2500が発売された。オリジナルに忠実に移植され，しかもスピードは98版よりも速い。君もこの名作に挑戦しよう。

MZ-2500用　3.5D：7,800円
BPS　☎045(421)7421

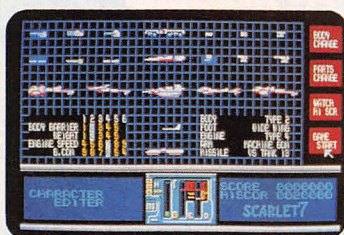
リグラス

森田和郎のランダムハウスが初のオリジナルブランドとして発表した3DリアルタイムRPGがこのリグラスだ。アルゴリズムを追求する森田氏の新技法により，フルカラー3次元の舞台が高速ヨコスクロールする。6段構成の舞台に秘密のアイテムも豊富。そしてレベルアップの果てに本当の謎解きが始まるのだ。

X1/X1turbo用　5D：6,800円
ランダムハウス　☎0298(42)1307

SCARLET 7

高速ヨコスクロール型シューティングアクション。コードネーム「Scarlet 7」という特殊コンテナを輸送するトランスポーターに乗り，USE軍の攻撃を突破しなければならない。ボディチェンジやパーツチェンジなどマニアックなメカシステムが魅力だ。

X1/X1turbo用　5D：5,800円
〃　T：3,800円
ソフトプロ　☎06(363)1221

ユーカラ

日本語ワードプロセッサとしてスタンダード的な位置を占めるユーカラのMZ-2500版。データの互換性や通信機能も信頼性がある。

MZ-2500用　3.5D：28,000円
東海クリエイト　☎03(456)4610

ぱれっと

日本語とイラストによるカラーグラフィック印刷キットで，POPやダイレクトメールなど幅広い活用ができるツールである。「ユーカラ」とのデータも利用できる。

MZ-2500用　3.5D：30,000円
ダイナウェア　☎0727(62)2424

トップマネジメント

トップマネジメントは会社経営をシミュレート体験できるマネジメントゲームである。社内研修から個人の自己啓発用としても楽しく活用できる。

MZ-2500用　3.5D：19,800円
X1turbo用　5D：19,800円
光栄　☎044(61)6861

テラ

16ビットマシン用の日本語ワードプロセッサ「テラ」がX1turbo用として登場した。やさしさに多機能を加えてパーソナルビジネスに威力を発揮することだろう。

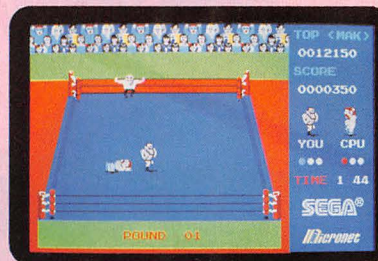
X1turbo用　5D：32,000円
日本マイコン販売　☎03(366)3274

インクポットデータライブラリ

Vol.4　パーティキット
Vol.5　クリスマスキット
Vol.6　年賀状キット
Vol.7　イラストアルバム2
Vol.8　挨拶状

X1turbo/PC-8800共用　5D：各3,800円
アスキー　☎03(486)7111

今月のGAME REVIEWから

ドルアーガの塔

チャンピオンプロレス Special

NOBO

軽井沢誘拐案内

JOY JOY PACK SPECIAL
ソフトとボードで2重にワイワイ

MZ-1500ユーザーのみなさん，クリスマスプレゼントの予約は済んだかな。また，もう冬休みでワクワクのみなさん，お正月ゲームの準備は万全かな。では，まだこれからという人を思いっきりひいきして，このJOY JOY PACK SPECIALを紹介するとしよう。

家族みんなで楽しめるソフトをパックしたJOY JOY PACKシリーズの第3弾が発売された。今までのような，ゲームあり，ワープロあり，といった幕の内タイプと趣向を変え，パック全体がユニークなひとつのゲームという構成になっている。

さて，クリスマスやお正月ともなれば，どこの家でもボードゲームのひとつやふたつは遊んだ経験があるだろう。最近ではボード版のロールプレイングゲームなども人気があり，ドルアーガの塔などもボードゲームが作られて好評だ。というわけで，今回のJOY JOY PACK SPECIALはご覧のような楽しいボード盤が大きなポイントになっている。では，全体の内容を見てみよう。

まず，メインゲームとなるのはドロールというゲームソフトだ。これは，ロードランナーなどでお馴染みの米国ブローダーバンド社の人気ソフトで今回MZ-1500のために移植されたものだ。これに，シューティングゲームとサイコロの役割やゲーム進行のためのソフトが2枚のQDにまとめられているのだ。

JOY JOY PACK SPECIAL
MZ-1500用　QD 2 枚組　9,800円
SBCソフトウェア　☎03(353)9241

シューティングゲーム

ドロール

それでは，実際のゲームについて簡単に説明しよう。

まず，メインゲームとなるドロールだが，宇宙大魔人にさらわれたお母さんや弟，妹を助け出すため特殊スーツに身を包んだドロール君が4階建ての地下牢に挑むスリル満点のアクションゲームである。このJOY JOY PACKでは，ドロール君の家族救出の物語を宇宙的スケールにふくらませている。

物語は，ドロール君が家族救出の準備のためにアルバイトをしてお金をためる地球編に始まる。お金がたまったらいざ宇宙へ出発というわけだ。宇宙にはいたるところで戦闘エリアに突入してしまい，ここでシューティングゲームをクリアしなければならないのだ。このシューティングゲームも単独のパソコンゲームとして遊べるが，難易度が2段階あり，さらに宇宙船カードや途中で手に入れた武器カードのキーワードを入力することによってゲーム展開が有利になったりするのだ。そのほか，戦闘エリアをワープできることもあれば，ブラックホールに落ちて脱落ということもある。こうして数々の難関を突破したドロール君を待ち受けているのがメインゲームのドロールである。1面は妹，そして弟，最後にお母さんを救出すれば感動的なデモ画面となり，助け出した家族の分だけボーナス点となるわけだ。

このように，このJOY JOY PACK SPECIALでは，ドロールやシューティングゲームなどのパソコンゲームを有機的に結びつけて全体をひとつのボードゲーム（ようするにスゴロクだ）にまとめあげており，数人でプレイすれば，ひとりでゲームソフトをプレイする何倍もの楽しみが味わえるようになっている。もちろんゲームを盛り上げるためのイラストボードやお金，約束手形，進行用ソフトも付いて楽しさいっぱい。この冬はJOY JOY PACKで思いっきりワイワイやろう。

パソコンは格闘技だ

清水和人のゲームハイテク道場

さあゲームも続々と出たところでよくプレイしたゲームにけりをつけようじゃないか。時代は今や完璧にロールプレイングしとる。続編だのいろいろあとがつかえているけん，3本＋αをまとめて面倒みようというわけだ。ページもないのでさっさといってみよう！

ファンタジアン

ファンタジアン

まずは死ぬほど遊んだファンタジアンから。

❶アイテム情報

クラスチェンジ：Staff of Sage, Helm of Aslam, Ring of Gold, Crystal Idol

ストーリー用：Crest of Gold, Dragon's Scale, Glocial Crystal（これらはそれぞれの場所に2度行くんだったかな？）

ハテナ編：Golden Crossはスケルトン，ゾンビ，ゴーストなどにのみ有効。Key of Chestこれは合い鍵

武器：Irom Staff（パワー50），Fire Cross（75），Whip（75），Thunder Axe（250），Sword of Evil（300），Silver Sword（400），Mum blade（？），

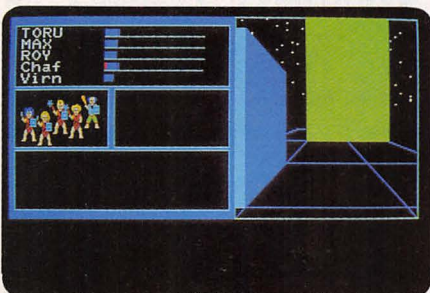
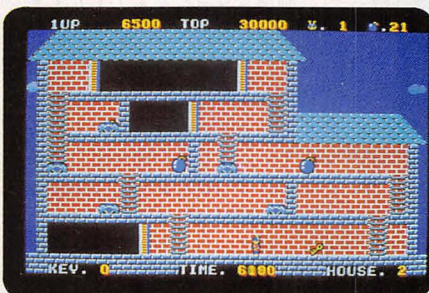
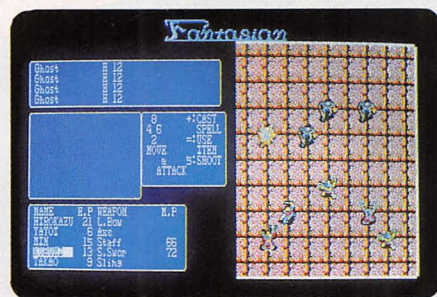
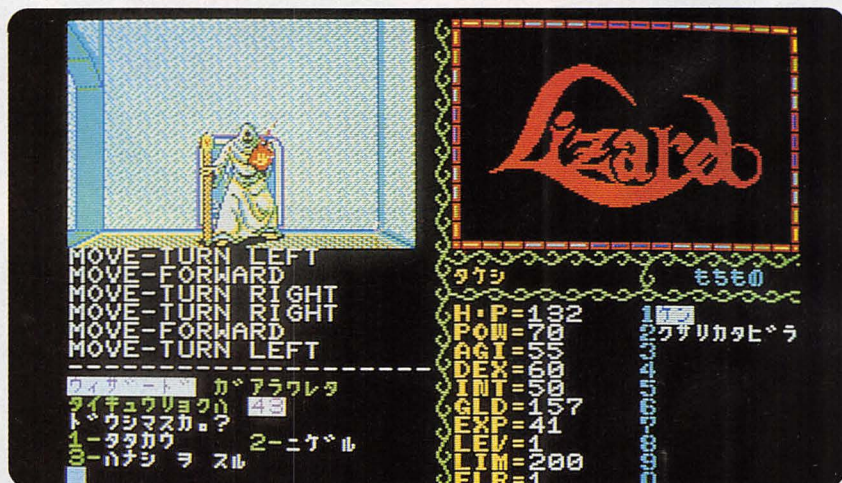
防具：Prayer Shisld（防御力1，ただしATTACK SPELLの被害を小さくする。）Winged Helm(2), Golden Groves(3), Silver Groves(2), Silver Plate(7), Elven Chain(5), Holy Robe(4), Darable Chain(R)

❷戦闘テクニック

ATTACK SPELLをさけるために，敵に密着する。ATTACKやBITEの強い敵にはLong Bowで。

❸キャラ作り

・ボーナスポイント9まで何度もやり直す。

・ステータスはDEXを大きくして攻撃の確度を増す。

リザード

・5人作ってJoin Goldして4人コロス。

・戦士（ドワーフ），盗族（ハーフオーク），修道士（ハーフリング），魔術師（ハーフエルフorエルフ）

❹装　備

初期状態

Battler：L. Bow, L. Shield, Chain, Helm

Thief：Axe, S. Shield, Leather

Illusionist：S. Sword, S. Shield, Chain

Monk：Staff, Leather

最終状態

Fighter：Mum Blade, Silver Plate, Winged Helm, Prayer Shield, Golden Groves

Robber：Sword, Durable Chain, Helm, Prayer Shield

Magician：Irom Staff, HolyRobe, Prayer Shield

❺マル秘作戦

・EQUIPで＃マークの付いているやつにアイテムをあげると，アイテムが2つに増える。

・Fighter以外は薬をひとつ多く持てる。

・アイテムは4階の先の泉か，5階のResting Roomで出やすい。

❻キャラ育て

地下4階のレベルダウンの泉でレベルを下げてはまた旅に出る。いつかきっと各パラメータ(DEX, STR, VIT, AG, WIS, INT, HITS)がよく上昇する。そうしたら次のレベルでもう1回同じことをする。泉でレベルは下がってもパラメータは下がらない。HITS以外のパラメータは18まで上がる。また，クラスチェンジの前は必ずセーブしておき，各項目があまり上がらなかったら，やり直すという方法もある。

ザ・ブラックオニキス

デーモンクリスタル

ザ・ブラックオニキス

以前かなり詳しい解説が本誌にも載ったが，未だに質問が来るソフト。でもどうなんだろう。透明になる魔法のマントとスバラシイBattle Axeをもち，オニキス（地上

2階のマップさえ書ければ簡単に見つかる）を見つけまくればそれでいいのではないか。他の階はマップなんかいらないよ（でも地下1階と，地下6階のマップはいるかもしれない）。「イロイッカイズツ」も単純なひっかけなのだ。X1では逆順であり，しかもドアの色は無視するが，降りた場所の黄色を抜かしてはいけないのだ。つまり最初はああ行ってすぐ戻るのが正解ってわけさ。

さらにオニキスを何度も発見してレベルを上げる方法があるらしい。それは先頭の人がからんでいるとかいないとか。そして私は今モーレツに怒っている。「ファイヤークリスタル」がテープ版だけ出たのだ。俺は「オニキス」をディスク版でやってたんだぞー。ファンタジアンならテープ版のキャラをディスク版に移せるんだぞお。アイテムだって多いしファンタジアンのほうが上だい！　まあオニキスのほうがスピードが速いけどね。

デーモンクリスタル

先月号で紹介したがまだ不十分だったようなので付け加えておこうっと。みんなは30面でとっても苦労しているらしくて，「こうやれば解ける」というハガキの中にも複雑すぎるのがあるんだよね。1階と3階で左右の××に×当た×をしてドアを××っぱなしにして，TIMEが7000を切ると3つの玉が出ましたね？　そこでそれをブラックオニキスのように取るのです。つまり順番があるのだよ。しかもよく知っている順番だ。すると敵が来るのでビシバシとやっつけるよろしある。ビシバシとだよ。先月の隠しコマンドを知っていれば誰でもクリアできるよね(MZ-1500用はまだよく知らんけど)。

リザード

さあ今回目いっぱい遊んだリザードの紹介だ。これはファンタジアンの「クリスタルソフト」から出ているなかなかのRPGである(でも××××クロスはでえっ嫌いだ)。

パーティを組むのではなく，ひとりで寂しく行くのだ。ファンタジアンと同様，アイテムがいっぱい出てくるし，戦闘モードも，敵キャラの絵もなかなか凝っていて楽しい。

① キャラ作り

戦士か魔術師か盗賊で，各項目は職業によって一定である。お金もファンタジアンのように金持ちから出発できないので，始めの装備は貧弱である。例によって商店，寺院，病院，宿屋もある。で地下の迷宮ではなく地上10階建てという豪華な屋敷である。

② マップ作り

なんとマップはヒントである程度ついてくるのだ。あとは一方向ドア，宝箱の位置，敵が必ずいる場所などを記入していけばよい。もう方眼紙はいらない。上下へのハシゴではひとマスのところにあることが多い。

③ 戦闘

ひたすらタイミング，すなわちアクションゲームである。戦闘のシーンで敵と自分が動くのだが，敵がうしろへちょっと下がったところを攻撃すると成功の確率が高いようだ。また1度攻撃が成功したらテンポよく2度3度とやっつけよう。自分がころんだ直後も有効だし戦闘開始の音楽が鳴り終わった直後も成功しやすい。しかしやはり強い相手は強いのでどんどんこちらもレベルアップしたい。危ないなと思ったらすぐ帰って治療することだ。セーブも欠かさないようにしたれ。

④ 商店

ここでは比較的早く最高金額の武器や防具が買えるようになるが，あまり不相応なものを付けていると，壊れてしまうことが多い。そんなに高くなくても十分通用するから無駄遣いしないように。また買ったものは装備し忘れないよう気を付けて。

武器の種類

タンケン	Gold20	POW＝＋5
ギンノタンケン	Gold40	POW＝＋6
コンボウ	Gold50	POW＝＋6
ヤリ	Gold80	POW＝＋10
オノ	Gold100	POW＝＋12
セントウオノ	Gold150	POW＝＋15
ケン	Gold200	POW＝＋20
ホシノケン	Gold300	POW＝＋22
ツキノケン	Gold500	POW＝＋25
タイヨウノケン	Gold1000	POW＝＋30

防具

カワノローブ	Gold50	AGI＝＋5
クサリカタビラ	Gold200	AGI＝＋10
テツノヨロイ	Gold300	AGI＝＋20
ドウノヨロイ	Gold200	AGI＝＋15
ロックプレート	Gold250	AGI＝＋18
ハードロックス	Gold400	AGI＝＋25
キノタテ	Gold50	AGI＝＋5
ドウノタテ	Gold100	AGI＝＋8
テツノタテ	Gold200	AGI＝＋10
ミュウノタテ	Gold1000	AGI＝＋16

⑤ 病院

ここがこのゲームのキーポイントである。病院では次の4つのモードがある。

・治療：1000goldで全快
・毒抜き：30goldで全快
・薬を買う
・LIMITを増やす：レベルに応じた金額でLIMITを増やせる。このゲームではレベルアップしてもLIMITは増えない。

このうち3番目の薬にはいろいろある。

(イ) キズグスリ	100gold	
(ロ) ゲドクザイ	60gold	
(ハ) ウデノクスリ	90gold	POW＝＋10
(ニ) アシノクスリ	80gold	AGI＝＋10
(ホ) アタマノクスリ	70gold	INT＝＋10

がある。しかし(ハ)(ニ)(ホ)は一時的にしか効かないのであまり使わない。薬よりレベルアップのほうが大切なのだ。

⑥ 宿屋

最初のうち少し使うかもしれないが，概して無用の場所。傷をいやすなら病院で治療するのがいちばん。

――――――――つづく――――――――

協力　栃木　周さん，神奈川　水井さん
宮崎　渡辺さん，宮城 満久保さん
有難うございました。記念品をお送りいたします。

GAME REVIEW

GAME REVIEWでは，最近発売されたゲームの中から，気になるソフト，見逃せないもの，読者の要望の高いものなどを対象としています。今月は，ドルアーガの塔，NOBO，チャンピオンプロレスSpecial，軽井沢誘拐案内の4作品です。ご意見ご希望などをお待ちしております。

ドルアーガの塔

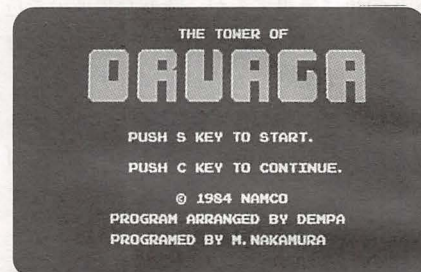
ファミコンで人気ナンバーワンのソフト，ドルアーガの塔がMZ-1500版で登場！　君もドルアーガを倒して，愛するカイを救いだせ。

THE TOWER OF DRUAGA。このタイトルの響きに底知れぬ魅力を感じるのは私だけであろうか。ドルアーガの塔は言わずと知れたナムコのビデオゲームで，この夏ファミコンに移植されて大ヒットとなった。パソコン用はMZ-1500版が初めての登場である。移植の出来映えからいうと，さすがにスピードや操作性にムリがあるようだ。それでも，グラフィックはなかなか美しく，音楽はちょっと軟弱な感じだが一応原曲に忠実に作られていて楽しい。

とにかく，ドルアーガフィーバーはかなりのもので，必勝本もベストセラーになるほどだ。ゲームセンターでなにもわからぬまま100円玉を浪費してしまった人は，ぜひ敵をとってもらいたい。というわけで，ギルはドルアーガを倒せるか？　N.N.

あのハイドライドでレディーアーマーを連続○人倒さないと○○しないとか，ウイザードの魔法を5発受けてから倒さないと××できないとかいうやつは，み～んなドルアーガのせいなんだゾと，ぼくは信じているのです。それほどドルアーガは苦しいゲームでした。今でこそ，いろいろな秘密が明かされ，解決の手引書なるものも出ているが，ほんとうに地道にひとりで謎を解き明かしていった人というのはどんなやつなんだろうと思わず身ぶるいしてしまう。

このMZ-1500用はなかなかの力作だけどキャラクタをスプライトのように動かすのはできないものかなあ。それでも，ちゃーんと裏ドルアーガまであったのには感心した。1500ユーザーにはおすすめです。

P.S.トレーナーも買っちまったい！　S.S.

H.K.							評価項目	S.S.						
					◀	◀	操作性	▶	▶	▶				
	◀	◀	◀	◀	◀	◀	グラフィック	▶	▶	▶	▶	▶	▶	
	◀	◀	◀	◀	◀	◀	サウンド	▶	▶	▶	▶			
			◀	◀	◀	◀	アイデア	▶	▶	▶	▶			
			◀	◀	◀	◀	熱中度	▶	▶	▶	▶	▶		

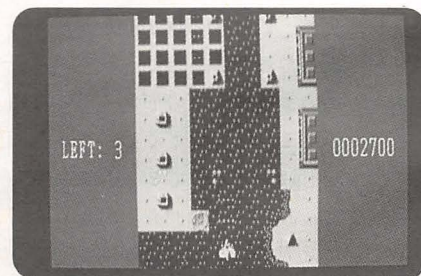
MZ-1500用　QD：4,800円
MZ-2500用も発売予定
電波新聞社　☎03(445)6111

NOBO

MZ-2500のスムーズスクロールが生かされたゼビウスタイプのシューティングゲーム。1周29分の戦闘空域には壮大なシナリオが隠されている。

味方機は戦闘機然としていて翼もあるし，敵機は丸や四角のシュールな形だし，回転板はあるしピラミッドもあるという典型的なゼビウス型シューティングゲームである。確かに評判のスクロールは滑らかだし，オープニングミュージックも軽快で良いが，画面両サイドの情報が少なく，戦闘中のBGMもないに等しいので退屈さを感じる人もいるだろう。などとケチをつけたわりには10分の壁が破れず，私は泣いている。うーん。何分後にどのエリアに達するかがわかるのは憎い。どうせなら，戦闘中に時間もわかればいいのに。

最後に，どの敵がどう動くかはすぐに把握できるので，それをいかに見切るかがポイントだ。そしてプレイヤーは，29分の死闘に耐えねばならない。　K.Y.

これはグラフィックにものを言わせたゲームです。ご存知のようにMZ-2500版のスクロールは極上の滑らかさで見るものを圧倒させます。このNOBOは1周するのに29分間もかかるという壮大なるゲームで，敵の攻撃は休む間もなく続きます。これを連続プレイで楽しみましょう（ESCキーは外しておきましょうね)。ゲームを進めていくと隠れキャラクターを思わせるものが画面の中に登場してきます。120エリアをクリアするのは困難でしょう。自然の照準がないのは残念ですが（ジェルダもそうでした)，慣れることですね，はっはっは。気が付いたら殺されていたというのはグラフィックのせいです。そう，君は雨あられと降る弾の合い間をぬううちに，「弾が止まって見える」と言うことだろう。　R.S.

K.Y.							評価項目	R.S.						
		◀	◀	◀	◀	◀	操作性	▶	▶	▶	▶			
		◀	◀	◀	◀	◀	グラフィック	▶	▶	▶	▶	▶		
			◀	◀	◀	◀	サウンド	▶	▶	▶	▶			
				◀	◀	◀	アイデア	▶	▶	▶				
			◀	◀	◀	◀	熱中度	▶	▶	▶	▶	▶		

MZ-2500用　3.5D 6,800円
コムパック　☎03(375)3401

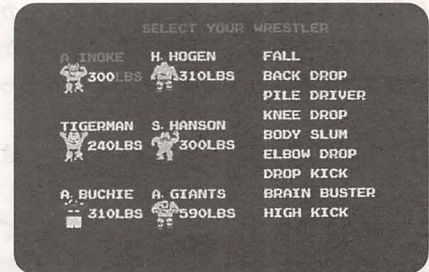
評価段階			
▶▶▶▶▶▶▶▶	素晴しい	▶▶▶▶	普通
▶▶▶▶▶▶▶	よく出来ている	▶▶▶	少しもの足りない
▶▶▶▶▶▶	まずまずである	▶▶	劣っている
		▶	劣悪

評価グループ
有田隆也　浅野恵造　祝一平　牛嶋昌和　工藤誠　挙市哲司　こうもと
やすひこ　小森隆　近藤弘幸　斎藤晋　斎藤亮　佐藤友彦　清水和人
白河哲　立花かおる　中川智哉　Hiroshi Omaeda　茗原秀幸　吉田幸一

チャンピオンプロレスSpecial

MZ, XIにとってまさに待望のプロレスゲームが登場した。人気レスラー総出演でエンタテイメントとしてのプロレスが楽しめる。

やっぱり，プロレスはこうでなくっちゃあ。ちまちました関節技よりもラフファイト，大技の連発で“見せるプロレス”をパソコン上にそのまま再現したようなソフトなのだ。キャラクタは6種類（テープ版は2種類）あって，どれもかわいいと言えるほどコミカルなのだが，試合の内容は大真面目である。絶対見ることのないA.ジャイアン○のパイルドライバーや，近頃見かけないブッ○ーの凶器攻撃が見れるんだから，プロレスファンにとってはこたえられないだろう。場外乱闘で鉄柱攻撃，フェンス直撃も自由自在。ただし，3分3本勝負なので，ぐずぐずしていると時間切れで相手（チャンピオン）の防衛となってしまう。これでキャラクタのコンストラクションがあれば完璧なんだがなあ。　M.K.

場内，興奮に包まれております。タイガー対ジャイアンツの一戦。タイガーが体重差をどうカバーするのでありましょうか，注目されます。3分間3本勝負の短期決戦。さあ，ゴングが鳴った！　おーっと，いきなりドロップキック，返す刀でローリングソバットだ，これはたまらない，ジャイアンツ，もんどり打って倒れました。すかさず上からエルボーの嵐だ，カウントが入った。1，2，3，決まった。1本目は速攻でタイガー先取！　さて2本目，おっと今度はタイガーが捕まった。軽量の悲しさ，いいように振り回されている。どうやら2本目はジャイアンツ優勢の模様。しかし，誠に残念ながらここで放送時間がなくなってしまいました。うわっ場外乱闘だ，それでは皆さん，さようなら。　T.K.

M.K.	評価項目	T.K.
◀◀◀◀	操作性	▶▶▶▶
◀◀◀◀◀	グラフィック	▶▶▶▶▶
◀◀◀◀	サウンド	▶▶▶▶
◀◀◀◀◀	アイデア	▶▶▶▶
◀◀◀◀◀◀◀	熱中度	▶▶▶▶▶

MZ-2000/2200/2500用　T：4,800円
XI/XI turbo用　5D：6,800円
〃　T：4,800円
マイクロネット　☎011(561)1370

軽井沢誘拐案内

マンガやアニメからパソコン業界まで，めっきり有名人となった堀井雄二の最新作がこのゲーム。大胆な着想のニュータイプアドベンチャーだ。

ナウイ町，軽井沢のお話です。その昔，中軽井沢，北軽井沢と次々に繁殖して，そのうち東京の練馬界わいまで軽井沢という地名になってしまうんではないかと危惧された恐怖の町で，カワイコちゃんが突如として誘拐されてしまいます。

そして，まいどおなじみの犯人捜しとなるわけですが，あっちこっちで話を聞いて，抱きよせたり，しめあげたり，押したり，たたいたり，あの手この手で誘拐犯人を見つけようとしているうちに，思わず目的を忘れてしまい，テニスクラブでニタニタ，あさみちゃんとお話してホクホク，思わずホテルの支配人まで抱き寄せて大混乱，最後は神様仏様にまで協力してもらって，いざRPGへと突入するのでありました。いやー面白かった……。　T.S.

カルイザワッ　アソレ　カルイザワッ
買ったゲームは軽井ざわっ，買ったオイラも軽井ざわっ（以上東北民謡より）。

なんてえこったい。あのポートピア殺人事件を作った人がこんなになっちゃったぜ。これがなんとも面白い。しかしなんで最近甘口のソフトが多いんだろう。きっとみんなホイホイ買うのやろうなあ。しっかしまあそんなに難しいアドちゃんではない。出てくる画面が軽井ざわっじゃなくて，出てくる場面とストーリーも軽井ざわっじゃなくて，その両方を楽しむっていうアドベンチャーですな。ウワァなんてえ刺激的なんだってついつい引き込まれるわっけー。でもこんな過激なロードの仕方じゃディスクが壊われるんじゃないかと思うわっけー。日本語は正しく使いましょう。　K.S.

T.S.	評価項目	K.S.
◀◀◀◀◀◀	操作性	▶▶▶▶
◀◀◀◀	グラフィック	▶▶▶▶▶
◀◀◀	サウンド	▶▶▶
◀◀◀◀◀	アイ・デア	▶▶▶▶▶▶
◀◀◀◀◀◀	熱中度	▶▶▶▶▶

XI/XI turbo用　5D：5,800円
〃　T：4,800円
エニックス　☎03(366)4345

SPECIAL REVIEW
印刷工房 X1turboユーカラ用

この印刷工房を利用するとX1 turbo用のユーカラで作成された文書で多彩な印字が可能です。しかも，お手持ちのプリンタが8ピンや16ピンでも，24ドット明朝体文字に変換されて印刷できるというまったく不思議なソフトです。

最近は漢字プリンタ（24ドットのもの）もかなり手に入りやすくなってきました。しかし，8ドットのプリンタを使っている方も多いと思います。それがこの「印刷工房」を使うと何と8ドットのプリンタでも24ドットの明朝体文字が印刷できるようになるというのですから驚きです。もちろん，16ドットのプリンタでもできますからほとんどあらゆるプリンタが24ドットの漢字プリンタと同じになるということです。明朝体文字が印刷できるということだけでなく，アンダーライン，網かけ，罫線，斜体文字，4倍角や¼角文字も印字でき，分数表示やルビの印字もこなしてしまう（すべてのプリンタで可能）というすごいソフトです。

さて，このソフトはユーカラ（東海クリエイト社製ワープロソフト）で作成した文章を印刷するというものです。したがって文章の作成には当然ユーカラが必要です。いろいろなタイプの文字の指定はユーカラの文章作成中に行います。特殊な印字をしたい文字列の前に＿（アンダーバー）と英数記号（半角）の組み合わせで機能を指定します。こうして作成した文章を印刷工房で呼び出して印字するわけです。

豊富な印字機能

24ドットの明朝体で漢字が印字できるということのほかに具体的な機能を挙げてみましょう。

a)斜体文字：文字を斜めに傾けて（角度は2通り）印字する。
b)網かけ文字：文字に13種の網かけと2種類の訂正ラインを印字できる。
c)強調文字：文字の縦線を太く印字する。
d)反転文字：文字の白黒を反転させる。
e)淡調文字：文字を3段階に淡く印字する。
f)カラー印字：カラープリンタの場合に7色のカラー印字ができる。
g)縦倍角・4倍角：文字を縦方向に2倍にします。横も倍角ならば4倍となる。
h)接続改行：行と行をつなげる。
i)⅔改行：⅔行改行する。分数表示やルビに利用。

といったような豊富な印字が行えるほかにこれらの複数の組み合わせ（たとえば，4倍角で斜体字にして網かけをする）も可能です。

ただ，ひとつの機能を指定するのに全角文字1文字分（半角2つ）の場所をとるので，1行の文字数がその分減ってしまうという欠点があります。

印字サンプル　（ブラザーM-1024使用）

印刷工房　印字サンプル

ユーカラ と 印刷工房 を組み合わせると、いろいろの印字を楽しむ事ができます。
斜体文字 、 網かけ文字 、 字消しライン 、 強調文字 、 淡調文字 の他、
反転文字 などが自由自在にできます。
また、分数表示(ぶんすうひょうじ)やルビ打(う)ちに便利(べんり)な2/3 改行(かいぎょう)もできます。

$$\frac{X^2+2X}{3X^2+7X+4} \div \frac{X^2-4}{3X^2-2X-8} = \frac{X(X+2)}{(3X+4)(X+1)} \times \frac{(3X+4)(X-2)}{(X+2)(X-2)} = \frac{X}{X+1}$$ なんてこともできます。

外字もいろいろあります。

✂☎♡♣◇♤ …… AMPM

プリンタ制御と外字管理

文字種類が豊富なことに加えて印刷するときの機能も豊富です。具体的には，印字方向（縦・横），ページ印字，用紙サイズ，1行の文字数，文字・行間，袋綴じ，部数などの設定が行えます。また，アンダーラインは，127種類の組み合わせが選べます（ただし，1回の印刷には1通り）。ほかにも印字を開始するところと終了するところがページと行単位で指定できます。

外字は512種類が登録されていて変更もできます。作成のエディタは非常に強力で24ドットのフォントも楽に作れます。ほかの文字のフォントを呼び出したり，一部分を縮小したり，反転・回転などもできます。また，印刷工房で作成した外字の任意の30字をユーカラに転送できます（ユーカラのエディタは使いづらいが……）。

とにかく，凝った印字ができるので人に見せる文章の印字には最適です。クラス新聞やサークルの連絡，回覧板などに便利です。印字速度は遅いので大量の印字には向いていませんが，印刷の楽しみを満喫できるソフトです。

(north fox)

「印刷工房」(X1 turbo用，要ユーカラ)
5D2枚（マニュアル付）　14,000円
モーリン　☎03(457)7479

THE SENTINEL

共通I/Oポート，FM音源ボードはもう製作しましたか？　今月はMZ/X1がモノフォニックシンセサイザとして使えるFM音源サウンドエディタの発表です。ここでひと言。ヤマハのYM-2203, YM-3014が高すぎるという人。掲載した価格は定価(？)ですから，これよりも安く買えるはずです。また，実際に手に入りにくいという人もとりあえず1セットだけでも購入しましょう。FM音源ボードを製作するとき，ICソケットはすべて取り付けたうえで，左右いずれか一方に石をさしてください。もう一方のソケットはあけておけばよいのです。3和音までですが鳴らすことができます。なお，今月のサウンドエディタを使うときは部品配置図右列にYM-2203, 3014をさしてください。LFO(147ページ参照)は使えませんが，ほかは正常に動きます。

●Lisp入門はいい。最初S-OSのLispを打ち込んだとき「こんな苦労の代償に何かいいことあるのだろうか」と思っていた。入門を読むたびに苦労が報われてくる気がする。

山田　剛（14）X1Cs

しっかり読んで勉強したら，Lisp上の簡単なアプリケーションでも作って送ってみてください。この欄で紹介していきたいと思います。

●私としては貴誌に対して高度なコンピュータ雑誌としての道を進んでいただきたいと思っている。ついては共通モニタの言語アプリケーションとしてForth, Pascal, PL/Iなどの高級言語を制作し，さらにハードウェアについてもどんどん進めてほしい。

山田　昌良（17）MZ-700，MZ-2200

●誰か10進→2進変換ルーチン（浮動小数点用）と浮動小数点演算パッケージを作ってください。Lisp, Prolog, Forthなどが整数型で終わらないために！

関根　孝司（17）MZ-1500

とりあえずは整数型でもいいと思いますが，これが実現したらおもしろそうですね。テーマは「S-OSの言語に浮動小数点演算を！」。誰か挑戦しませんか？

●ZEDAを毎日便利に使わせてもらっています。皆さんもやっていることでしょうが，この前「Z80 ASSEMBLER～」と出力しているところをいじってメッセージを変えてみました。そのときにふと，そのあたりの表示関係をいじればおもしろいことができるのではと思い，いろいろ解析してみた結果，見事に成功しましたので報告します。

まず「Z80 ASSEMBLER～」のメッセージは1558行のところで出力しています（皆さんご存じでしょうね)。これをやめさせるには「CALL SPRNT」を「JP 3D52H」に変えます。

また，Z80のマシンコードは最長で4バイトですから，7バイトもオブジェクトを表示するのは無駄なように思います。これを変更するには1545行を書き換えればOKです。「IF A< >0 THEN LD B,3 ELSE LD B,7」の「LD B,3」はA//でアセンブルしたとき，「LD B,7」はA/でアセンブルしたときに表示されるオブジェクトのバイト数です。これを4にすれば4バイト表示になります。

これに合わせて行番号やソースの表示位置も前へずらしましょう。それには1505行に手を加えます。A//でアセンブルしたときには「LD B,15」が有効になり，A/のときには「LD B,28」が有効になります。4バイトのオブジェクト表示にするなら，4×3＋6で「LD B,18」，5バイトなら5×3＋6で「LD B,21」という具合に書き換えるとよいでしょう。

プリンタでページングするのを止めるには，1558行を「JP 3D58H」に変え，さらに1573行を「JP 3D62H」にすればできあがりです。

例として，A//でオブジェクトを4バイト表示にする方法を書いておきましょう。同時に「Z80～」のメッセージを削り，ページングの機能は残しておきます。私が現在愛用しているフォーマットです。

```
1558行 → JP 3D52H
1545行 → IF A< >0 THEN LD B,4
         ELSE LD B,7
1505行 → IF A< >0 THEN LD B,18
         ELSE LD B,28
```

オブジェクトを書き換えるなら，

```
3D0F  CD 22 38  →  C3 52 3D
3CF7  06 03  →  06 04
3C90  06 0F  →  06 12
```

となります。皆さんも自分の好みに応じてフォーマットを変えてみてはいかがでしょう。

田端　敏之（18）MZ-2000

皆さんの研究発表をお待ちしております。

●お願い。もう一度S-OS"MACE"の掲載を！僕は6月号が出た当時にはS-OS"MACE"の重要性がわかっていませんでした。しかし，7，8，9月号と次々にS-OS上で走るプログラムが掲載され，「うっこれはいいプログラムだ」と思ってもあとの祭り。"MACE"を持っていないので僕のマシンでは走らせられないのだ。6月号のバックナンバーは書店にもなく，友人も誰ひとりとして持っていません。6月号を買っていない僕にもS-OSを！

三木　隆浩（15）X1Cs

来月はいよいよ皆さんからのお便りをもとにパワーアップしたS-OSのバージョンアップ版を発表します。期待していてください。

●S-OS用にA.C.T.S.(アクツ)を作ろうと思ったがカタカナが使えないので断念した。しかたなくX1専用のA.C.T.S.を作ることにした。さて，A.C.T.S.とは何でしょう？

坂本　隆志（17）X1Ck

ACTって入ってるからアクションゲームか何かでしょうか。ところで，S-OSのバージョンアップ版ではカタカナもサポートします。さあ，これで作る気になったでしょう？

CIOS掲載記事

＊Lisp-85, ZEDA, ZAID, BEMS, ZING, MACINTO-S, CAP-X85, Prolog-85, FM音源サウンドエディタなどのアプリケーションプログラムは，基本オペレーティングシステムであるS-OS"MACE"(85年6月号)がないと動作しませんのでご注意ください。なお，2月号にディスク版を含むバージョンアップ版を掲載する予定ですので，そちらを利用することもできます。

全機種共通モニタCIOS

第13部
リロケータブルのお話

第14部
FM音源サウンドエディタ

リロケータブルのお話

リロケータブルなマシン語入力ツールMACINTO-Sの発表以来，プログラムを自由なアドレスに置くという話題は新鮮な反響を呼び，たくさんのお便りが寄せられています。今回はその中から２つの読者投稿を紹介しましょう。

チェックサムをリロケータブルに ALLOCATER

Hikita Takashi
疋田　孝

85年９月号にMACINTO-Sというツールが載っており，完全リロケータブルと書いてありました。この話に刺激された私は，前に手を入れていたプログラムをひっぱり出し，Oh!MZチェックサムプログラムにパッチ当てするプログラムを作りました。それが今回のアロケータです。このアロケータを使えば，S-OS用のOh!MZチェックサムプログラムが，リロケータブルなチェックサムに早変わりというわけです。

入力と起動方法

まず，Oh!MZチェックサムプログラムをロードします。

・B000$_H$～のチェックサムの場合

AF78$_H$よりリスト1-Aのとおりにダンプリストを入力し，AF78$_H$～B1FF$_H$を適当なファイル名でセーブします。

・3000$_H$～のチェックサムの場合

リスト1-Aを2F78$_H$より入力したあと，2F7D$_H$のAF$_H$を2F$_H$に変更し，2F78$_H$～31FF$_H$をセーブします。

好きな番地にロードして，そこへＪ命令でジャンプすると起動します。

〈注意〉

MACINTO-Sと同じジャンパを利用していますので，起動前に1F80$_H$から２バイトがE1$_H$，E9$_H$となっていなければいけません。

プログラムの説明

このプログラムは，ほかのアドレスに動かすと支障をきたすため変更しなければいけないデータに，テーブルを利用してパッチを当てるものです。

テーブルにはパッチを当てる個所が相対値として入っています。次のパッチ当ては何バイト先という具合です。このテーブルを自分で作りTSIZEを変えてやれば，ほとんどのプログラムはリロケータブルになります。

ただし，次にパッチを当てる番地の相対アドレスがFF$_H$をこえる個所がある場合はこのバージョンではサポートしていません。

リスト1-A　ALLOCATERダンプリスト

```
AF78 CD 80 1F 01 7B AF 54 5D :48
AF80 23 73 23 72 21 85 00 19 :EA
AF88 E5 21 00 00 39 EB F9 21 :44
AF90 3E 00 39 73 23 72 21 00 :A0
AF98 00 39 AF ED 42 44 4D 11 :B9
AFA0 85 00 21 41 00 39 EB 39 :44
AFA8 F9 1A B7 28 0B 13 26 00 :36
AFB0 6F 39 F9 E1 09 E5 18 F1 :79
AFB8 31 00 00 C9 02 03 05 06 :0A
AFC0 04 03 03 05 06 03 04 08 :24
AFC8 05 03 06 04 08 16 03 04 :37
AFD0 05 0A 06 04 0B 05 03 03 :2F
AFD8 03 03 03 03 07 0C 03 05 :27
AFE0 05 03 03 03 03 03 04 03 :1B
AFE8 04 04 06 03 09 0A 13 06 :3D
AFF0 03 03 0A 06 03 08 06 03 :2A
--------------------------------
SUM: 4E BD 20 02 7F 48 13 F8 :FF

AFF8 08 20 03 51 03 03 03 00 :85
--------------------------------
SUM: 08 20 03 51 03 03 03 00 :85
```

リスト1-B　ALLOCATERソースリスト

```
0000                 1 ;
0000                 2 ;  ***  ALLOCATER V.3-1  ***
0000                 3 ;
0000                 4 ;         BY  T.HIKITA
0000                 5
0000                 6 GETPC:  EQU     1F80H        ; POP HL,JP (HL)
0000                 7 EOT:    EQU     0            ; END MARK
0000                 8 PRGST:  EQU     0B000H       ; PRGST
0000                 9 ALSIZ:  EQU     0044H        ; ALLOCATER SIZE
0000                10 TSIZE:  EQU     0044H        ; TABLE SIZE
0000                11 START:  EQU     PRGST-TSIZE-ALSIZ
0000                12
AF78                13         ORG     START
AF78                14
AF78                15 ;  +++ GET PRGRAM COUNTER +++
AF78                16
AF78 CD 80 1F       17         CALL    GETPC
AF7B                18
AF7B 01 7B AF       19 PC:     LD      BC,PC
AF7E 54             20         LD      D,H          ; DE=PC
AF7F 5D             21         LD      E,L
AF80                22
AF80                23 ;  +++ FOR RESTART +++
AF80                24
AF80 23             25         INC     HL
AF81 73             26         LD      (HL),E
AF82 23             27         INC     HL
AF83 72             28         LD      (HL),D
AF84                29
AF84                30 ;  +++ SAVE RETURN ADDRESS +++
AF84                31
AF84 21 85 00       32 SAVE:   LD      HL,PRGST-PC
AF87 19             33         ADD     HL,DE        ; HL=PRGST
AF88 E5             34         PUSH    HL           ; SAVE IT IN STACK
AF89                35
AF89                36 ;  +++ SAVE SP +++
AF89                37
AF89 21 00 00       38         LD      HL,0
AF8C 39             39         ADD     HL,SP        ; HL=SP
AF8D EB             40         EX      DE,HL        ; HL=PC,DE=SP
AF8E F9             41         LD      SP,HL        ; SP=PC
AF8F 21 3E 00       42         LD      HL,SAVSP-PC
AF92 39             43         ADD     HL,SP        ; HL=SAVSP
AF93 73             44         LD      (HL),E       ; SAVE SP IN (SAVSP)
AF94 23             45         INC     HL
AF95 72             46         LD      (HL),D
AF96                47
AF96                48 ;  +++ CALCULATE LOAD OFFSET +++
AF96                49
AF96 21 00 00       50         LD      HL,0         ; INIT HL
AF99 39             51         ADD     HL,SP        ; HL=PC
AF9A AF             52         XOR     A
AF9B ED 42          53         SBC     HL,BC        ; HL=BIAS
AF9D 44             54         LD      B,H          ; BC=BIAS
AF9E 4D             55         LD      C,L
AF9F                56
AF9F                57 ;  +++ SET REGISTERS +++
AF9F                58
AF9F 11 85 00       59         LD      DE,PRGST-PC
AFA2 21 41 00       60         LD      HL,TBL-PC
AFA5 39             61         ADD     HL,SP        ; HL=TBL
AFA6 EB             62         EX      DE,HL
AFA7 39             63         ADD     HL,SP        ; HL=PRGST,DE=TBL
AFA8                64
AFA8                65 ;  *** 1 BYTE TABLE VERSION ***
AFA8                66 ;
AFA8                67 ;  CAUTION: OFFSET MUST BE LESS 100H
AFA8                68
AFA8 F9             69         LD      SP,HL
AFA9                70
AFA9 1A             71 CHG:    LD      A,(DE)
AFAA B7             72         OR      A
AFAB 28 0B          73         JR      Z,OUT
AFAD 13             74         INC     DE
AFAE 26 00          75         LD      H,0
AFB0 6F             76         LD      L,A
AFB1 39             77         ADD     HL,SP
AFB2 F9             78         LD      SP,HL
AFB3 E1             79         POP     HL
```

```
AFB4 09              80          ADD     HL,BC
AFB5 E5              81          PUSH    HL
AFB6 18 F1           82          JR      CHG
AFB8                 83
AFB8 31              84 OUT:     DEFB    31H          ; LD SP,XXXX
AFB9 00 00           85 SAVSP:   DEFW    0
AFBB C9              86          RET                   ; JP PRGST
AFBC                 87
AFBC                 88 ;  TBL
AFBC                 89 ;
AFBC                 90 ;    END MARK:00H
AFBC                 91
AFBC                 92 TBL:
AFBC 02 03 05 06     93       DEFB    0B002H-0B000H:005H-002H:00AH-005H:010H-00AH
AFC0 04 03 03 05     94       DEFB    014H-010H:017H-014H:01AH-017H:01FH-01AH
AFC4 06 03 04 08     95       DEFB    025H-01FH:028H-025H:02CH-028H:034H-02CH
AFC8 05 03 06 04     96       DEFB    039H-034H:03CH-039H:042H-03CH:046H-042H
AFCC 08 16 03 04     97       DEFB    04EH-046H:064H-04EH:067H-064H:06BH-067H
AFD0 05 0A 06 04     98       DEFB    070H-06BH:07AH-070H:080H-07AH:084H-080H
AFD4 0B 05 03 03     99       DEFB    08FH-084H:094H-08FH:097H-094H:09AH-097H
AFD8 03 03 03 03    100       DEFB    09DH-09AH:0A0H-09DH:0A3H-0A0H:0A6H-0A3H
AFDC 07 0C 03 05    101       DEFB    0ADH-0A6H:0B9H-0ADH:0BCH-0B9H:0C1H-0BCH
AFE0 05 03 03 03    102       DEFB    0C6H-0C1H:0C9H-0C6H:0CCH-0C9H:0CFH-0CCH
AFE4 03 03 04 03    103       DEFB    0D2H-0CFH:0D5H-0D2H:0D9H-0D5H:0DCH-0D9H
AFE8 04 04 06 03    104       DEFB    0E0H-0DCH:0E4H-0E0H:0EAH-0E4H:0EDH-0EAH
AFEC 09 0A 13 06    105       DEFB    0F6H-0EDH:100H-0F6H:113H-100H:119H-113H
AFF0 03 03 0A 06    106       DEFB    11CH-119H:11FH-11CH:129H-11FH:12FH-129H
AFF4 03 08 06 03    107       DEFB    132H-12FH:13AH-132H:140H-13AH:143H-140H
AFF8 08 20 03 51    108       DEFB    14BH-143H:16BH-14BH:16EH-16BH:1BFH-16EH
AFFC 03 03 03       109       DEFB    1C2H-1BFH:1C5H-1C2H:1C8H-1C5H
AFFF 00             110       DEFB    EOT ; END MARK
```

究極のリロケータブルシステム BOUNCER

Kita Akira
北　明

85年6月号から始まったS-OS，毎号システム系の発表が続き，編集室の並々ならぬ力の入れようにマイコンを持たぬ私はため息混じりに見ておりましたところ，9月号でリロケータブルについて取り扱っておりましたので便乗的に口をはさましていただこうというあつかましい所存であります。

リロケータブルについて

リロケータブルなプログラミングとZ80CPUとの闘いは今に始まったものではありません。おそらくはMZ-80Kが世に出た時代にはすでに始まっていたでしょう。しかし，その手段はというと，

1)　プログラム内の絶対アドレスに関する命令を探し出し書き換える。
2)　プログラム内の書き換えるべきアドレステーブルを用意し，それに従って書き換える。
3)　PCの値を得ることによって，絶対アドレスを修正しながら実行する。
4)　リロケータブルな命令のみ使い，絶対アドレスに関するものを使用しない。

のうち，もっとも消極的な4)に頼ったものが大部分でした。

以下4つの方法を見ていきましょう。

1)の方法はプログラム中のC3$_H$やCD$_H$などを探し出し，続く2バイトを書き直しておいてから実行するというものです。セルフ・リロケータブル・デバッガ(I/O)はこの方法を用いた代表的なものであるといえます。しかし，データ中にC3$_H$やCD$_H$があると誤って書き換えられる恐れがあるため注意を要します。

2)の方法は，あらかじめ書き換えるべきアドレステーブルを内部に用意することによって1)の方法の欠点を補おうとするものです。しかし，テーブルの製作・管理・変更がかなり煩雑であり，結果としてプログラムの変更が困難となってしまいます。結局，2)の方法によるプログラミングは非生産的と言わざるをえないでしょう。

3)の方法には多くのバリエーションがあり，MACINTO-Sに用いられたものもそのひとつです。この方法は『インターフェース』'82年2月号で初めてマイコン雑誌上に発表され，Z80CPU上で同じ目的を持って闘っていた多くの同志の注目と賞賛を浴びました。当時，ANCHOR(Oh!MZではジャンパと呼ばれている）として発表された方法が偶然にも3年前と同じ姿で使用されたことからももっとも有効的な方法と考えられます。

4)の方法はもっともポピュラーなもので，その例はあちこちに見受けられます。

BOUNCER登場

前おきが長くなりましたが，いよいよ本稿の真打ちの登場です。前述のANCHORと同時に発表されたBOUNCERにあやかって同じ名前を付けました。混同しては困るので，本稿では以下“先代”と呼ぶことにします。

このプログラムは，以下の点で先代と異なっています。

1)　LD命令を含むすべての絶対アドレス関係の命令をサポート。
2)　それに伴いプログラムサイズが大きくなった。
3)　同時に実行速度が多少低下した。
4)　BOUNCERを経由する前後において，レジスタ，フラグは一切影響を受けない。
5)　使用方法が簡単であるため，アセンブラを使用すれば，プログラムを固定アドレスで作ろうとリロケータブルに作ろうと手間がかわらない。

当然のことながら，BOUNCER自身はプログラムなどによって破壊されない安全な絶対アドレスに配置せねばなりません。

なぜリロケータブルにならないか

マシン語プログラムが通常リロケータブルでないのは，LD, JP, CALLの3つのコマンドのせいです。「JP (HL)」を除いて，JP, CALLは絶対アドレスにしか効きません。6000$_H$から始まるプログラム中に「CALL 6100$_H$」があるとしますね。このプログラムを8000$_H$にロードしても，「CALL 6100$_H$」は「CALL 6100$_H$」のままで，8100$_H$に変わってはくれません。JP命令も同様です。プログラムがないところへジャンプしたりコールしたりするため暴走してしまいます。当然といえば当然です。

思わぬ伏兵がLDです。「LD (nn),A」のnnも，ロードしたアドレスによって変わってくれたりしません。自分自身を壊してしまうことだってあります。(BC)，(DE)，(HL)に対するLDだって同じことです。

そこでBOUNCERの登場です。BOUNCERをかませることにより，上記の欠点を取り除きリロケータブルなプログラムを作ることが可能なのです。

BOUNCERの使い方

まずはLDへの対応ですが，これは図1のようにします。今までWORKというように絶対番地を書いていたところを，次の命令の頭との相対番地WORK-××にしておきます。すると，直前の「CALL BOUNCER」で，プログラムのアドレスに合わせて適正なアドレスに変換されるというわけです。図2を見てください。このようなアドレッシングにも対応できます。

JP命令には図3のようにして対応します。ここでもやはり，ジャンプ先は相対アドレ

スにしておきます。

最後にCALLですが，もう手法はおわかりでしょう。図4です。これは，Aが0のときトライする方法ですが，BOUNCERの前後でレジスタは保存されるため，ただ「CALL BOUNCER」を挿入しただけの形になっているところに注目してください。

このようにBOUNCERは使い方が非常に簡単なプログラムです。これだけでプログラムはリロケータブルになるのです。

リストはとりあえず8000Hでアセンブルしておきました。おもしろそうだなと思った方は試してみてください。

その際，ソースを打ち込み，自分の好きなアドレスにBOUNCERを置いて使ったほうが有効かと思います。

図1

```
        LD      (WORK),A
                 ⇓
        CALL    BOUNCER
        LD      (WORK-XX),A
XX：            ～
WORK：DEFB      0
```

図2

```
        LD      HL,WORK
        LD      (HL),B
                 ⇓
        CALL    BOUNCER
        LD      HL,WORK-XX
XX：    LD      (HL),B
                ～
WORK：DEFB      0
```

図3

```
        JP      STEP
                 ⇓
        CALL    BOUNCER
        JP      STEP-XX
XX：            ～
STEP：          ～
```

図4

```
        OR      A
        CALL    Z,RETRY
                 ⇓
        OR      A
        CALL    BOUNCER
        CALL    Z,RETRY-XX
XX：            ～
RETRY：         ～
```

おもしろいアイデアですね。ただし，作者も述べていらっしゃるように速度はかなり遅くなってしまいます。やはりZ80においてリロケータブルに，というのは邪道なのかもしれませんね。

すでにGETPCと[HL]がS-OSにあり，これ以上リロケータブルプログラムのためのルーチンは必要なしとの判断から，BOUNCERをS-OSに組み込むのは見送らせていただきます。（編集室）

リスト2-A　BOUNCERダンプリスト

```
8000 22 63 80 21 00 00 22 65 :AD
8008 80 22 67 80 21 69 80 36 :C9
8010 C3 E1 22 6A 80 ED 73 60 :70
8018 80 31 A0 80 F5 D5 7E 5F :78
8020 23 FE ED 28 47 E6 DF FE :40
8028 DD 28 4C 53 1E 00 7A FE :3A
8030 CD 28 4F FE C3 28 17 E6 :2A
8038 C7 FE C4 28 47 FE C2 28 :E0
8040 0D 7A E6 CF 3D 28 07 7A :22
8048 E6 E7 FE 22 20 0F ED 53 :5C
8050 65 80 5E 23 56 23 22 6A :6B
8058 80 19 22 67 80 D1 F1 31 :95
8060 00 00 21 00 00 00 00 00 :21
8068 00 00 00 00 7E 57 E6 C7 :82
8070 FE 43 20 E9 23 18 D7 7E :DA
8078 57 E6 F0 FE 20 20 DE 23 :6C
--------------------------------
SUM: A6 06 8A 8E F9 F1 67 34 :49

8080 18 BF 16 C5 7A D6 02 57 :5B
8088 ED 7B 60 80 23 23 E5 2B :9E
8090 2B ED 73 60 80 3E C9 32 :A4
8098 69 80 18 B2 00 00 00 00 :B3
--------------------------------
SUM: 99 A7 01 57 1D 37 B0 B4 :50
```

リスト2-B　BOUNCERソースリスト

```
0000                1 ;
0000                2 ;  BOUNCER VERSION 1.4
0000                3 ;
0000                4 ;
0000                5
8000                6          ORG     8000H
8000                7 ;
8000 22 63 80       8 BOUNCER:LD       (LDHL+1),HL
8003                9 ;
8003 21 00 00      10          LD      HL,0
8006 22 65 80      11          LD      (INST),HL
8009 22 67 80      12          LD      (ADR),HL
800C 21 69 80      13          LD      HL,RTN
800F 36 C3         14          LD      (HL),0C3H
8011 E1            15          POP     HL            ; PC
8012 22 6A 80      16          LD      (RTN+1),HL
8015               17 ;
8015 ED 73 60 80   18          LD      (LDSP+1),SP
8019 31 A4 80      19          LD      SP,SP18
801C F5            20          PUSH    AF
801D D5            21          PUSH    DE
801E               22 ;
801E 7E            23          LD      A,(HL)
801F 5F            24          LD      E,A
8020 23            25          INC     HL
8021               26 ;
8021 FE ED         27          CP      0EDH
8023 28 47         28          JR      Z,EXED
8025               29 ;
8025 E6 DF         30          AND     0DFH          ; 1101 1111B
8027 FE DD         31          CP      0DDH          ; 1101 1101B
8029 28 4C         32          JR      Z,INDEX
802B               33 ;
802B 53            34          LD      D,E
802C 1E 00         35          LD      E,0
802E 7A            36          LD      A,D
802F               37 ;
802F FE CD         38          CP      0CDH
8031 28 4F         39          JR      Z,CALLN
8033               40 ;
8033 FE C3         41          CP      0C3H
8035 28 17         42          JR      Z,COMSET
8037               43 ;
8037 E6 C7         44          AND     0C7H          ; 1100 0111B
8039 FE C4         45          CP      0C4H          ; 1100 0100B
803B 28 47         46          JR      Z,CALLC
803D               47 ;
803D FE C2         48          CP      0C2H          ; 1100 0010B
803F 28 0D         49          JR      Z,COMSET
8041               50 ;
8041 7A            51 BOUN1:   LD      A,D
8042 E6 CF         52          AND     0CFH          ; 1100 1111B
8044 3D            53          DEC     A             ; 01H
8045 28 07         54          JR      Z,COMSET
8047               55 ;
8047 7A            56          LD      A,D
8048 E6 E7         57          AND     0E7H          ; 1110 0111B
804A FE 22         58          CP      22H           ; 0010 0010B
804C 20 0F         59          JR      NZ,EXIT
804E               60 ;
804E ED 53 65 80   61 COMSET:  LD      (INST),DE     ; INSTRUCTION SET
8052 5E            62          LD      E,(HL)
8053 23            63          INC     HL
8054 56            64          LD      D,(HL)
8055 23            65          INC     HL
8056 22 6A 80      66          LD      (RTN+1),HL    ; RETURN ADDRESS SET
8059 19            67          ADD     HL,DE
805A 22 67 80      68          LD      (ADR),HL      ; RELATIVE ADDRESS SET
805D               69 ;
805D D1            70 EXIT:    POP     DE
805E F1            71          POP     AF
805F               72 ;
805F 31 00 00      73 LDSP:    LD      SP,0
8062 21 00 00      74 LDHL:    LD      HL,0
8065               75 ;
8065 00            76 INST:    NOP
8066 00            77          NOP
8067 00            78 ADR:     NOP
8068 00            79          NOP
8069               80 ;
8069 00            81 RTN:     NOP
806A 00            82          NOP
806B 00            83          NOP
806C               84 ;
806C 7E            85 EXED:    LD      A,(HL)
806D 57            86          LD      D,A
806E E6 C7         87          AND     0C7H          ; 1100 0111B
8070 FE 43         88          CP      43H           ; 0100 0011B
8072 20 E9         89          JR      NZ,EXIT
8074               90 ;
8074 23            91          INC     HL
8075 18 D7         92          JR      COMSET
8077               93 ;
8077 7E            94 INDEX:   LD      A,(HL)
8078 57            95          LD      D,A
8079 E6 F0         96          AND     0F0H          ; 1111 0000B
807B FE 20         97          CP      20H           ; 0010 0000B
807D 20 DE         98          JR      NZ,EXIT
807F               99 ;
807F 23           100          INC     HL
8080 18 BF        101          JR      BOUN1
8082              102 ;
8082 16 C5        103 CALLN:   LD      D,0C5H        ; C3H+02H
8084              104 ;
8084 7A           105 CALLC:   LD      A,D
8085 D6 02        106          SUB     2
8087 57           107          LD      D,A
8088              108 ;
8088 ED 7B 60 80  109          LD      SP,(LDSP+1)
808C 23           110          INC     HL
808D 23           111          INC     HL
808E E5           112          PUSH    HL            ; PUSH RETURN ADDRESS
808F 2B           113          DEC     HL
8090 2B           114          DEC     HL
8091 ED 73 60 80  115          LD      (LDSP+1),SP
8095 ED 7B A0 80  116          LD      SP,(SP14)
8099 3E C9        117          LD      A,0C9H
809B 32 69 80     118          LD      (RTN),A
809E 18 AE        119          JR      COMSET
80A0              120 ;
80A0 00 00 00 00  121 SP14:    DEFS    4             ; STACK
80A4              122 SP18:
```

全機種(共通I/Oポート, FM音源ボード, S-OS要)

音色作成&単音シンセサイザ

FM音源サウンドエディタ

Taga Masakazu
多画 正数

今回はBASIC各機種版とS-OS版のサウンドエディタを発表します。データは共通ですから，どちらで作った音色データも次に発表するミュージックプレイヤーで活用できます。MZ/X1がシンセサイザに早変わり。FM音源の多彩な音の秘密を探ってみましょう。

FM音源とPSG音源

なぜFM音源とPSGとでは音が違うのでしょう。それを知るにはまず「音の違い」とはどんなことなのかを知らなければなりません。それはだいたい次の4つの要因の複合作用によって生じます。

1) 音量
音の大きさ，ボリュームです。

2) 周波数
音の高さ，音程です。1)と2)はPSGでもコントロールできます。

3) エンベロープ
音量の時間的な変化を表したもので，たとえばピアノなら立ち上がりの速い"ポ〜ン"という音，ハーモニカならわりとゆっくりと立ち上がる"プゥ〜"という音になります。

4) 音色
これがいちばん説明しにくいものなのですが，ほとんどの人は耳で聴いてわかっていると思います。たとえば，同じ音程(高さ)の音でもピアノとバイオリン，トランペットなどエンベロープの違いはあるにしても明らかに音の感じが違います。これが音色です。

以上4つの中でFM音源とPSGで大きく異なるのはエンベロープと音色です。これらについて詳しく見ていきましょう。

エンベロープ

PSGにもエンベロープ回路はありますが，FM音源にはさらに細かいところまでコントロールするエンベロープ回路が付いています。PSGでは波形とそのサイクルしか指定できませんが，FM音源ではアタックタイム，ディケイタイム，サスティンレベル，サスティンタイム，リリースタイムと計5個のパラメータがあるのです(図1)。

これらはすべて音の鳴り方を示すもので，アタックタイムは音の鳴り始めから最大音量になるまでの時間，ディケイタイムは最大音量からサスティンレベルになるまでの時間です。サスティンレベルは，たとえばピアノなら鍵盤を叩いたときの打撃音の次の音（言葉で表すなら"ポォ〜ン"の"ォ〜ン"の部分)，すなわち響きの部分の音の大きさを表します。サスティンタイムはその音が続く時間，リリースタイムはキーを離したあとの音の余韻の時間となります。

これらをうまく組み合わせて設定することで，今までのPSGのような粗いエンベロープから，細かい音の鳴り方の部分までコントロールすることができます。

音色

音色というのは音程(高さ)，エンベロープ，音量が同じでも明らかに違う何かがあるために生じます。ひとことで言えば，それは波形の違いということになります。

ピアノ，バイオリン，トランペットなどあらゆる楽器の音には，基本となる周波数の音（基音）に加えて，その周波数が何倍かされた音（倍音）がたくさん混じっています。この倍音の混じり方，すなわち周波数成分が楽器によって違うのです。

音の基本がサイン波であることは，すでに学校で習っている方も多いことでしょう。したがって，周波数成分の違いとはサイン波の組み合わせ方の違いであり，グラフにしてみるとそれは波形の違いであるということができます。FM音源の最大の特長は

図1 エンベロープ

図2 PSGの波形

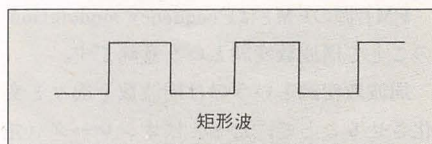
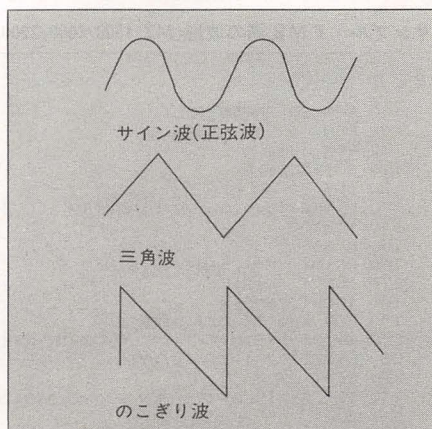
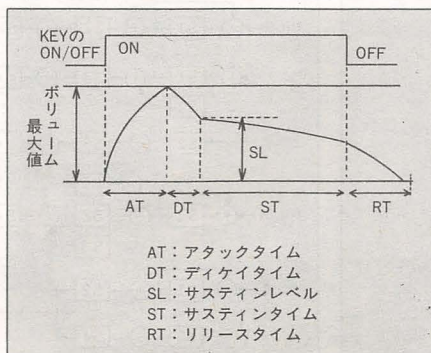



図3 いろいろな波形

この波形をコントロールできるということにあります。

まず図2を見てください。これがPSG音源により出てくる波形です。PSGは音色のコントロールはできませんので，波形はこれひとつだけです。しかし世の中の音にはいろいろな波形があります。たとえば図3のような波形では，周期が同じでも音色は異なります。PSGの矩形波なら“ポ〜”と鼻のつまったような音，サイン波なら音叉のような“ポ〜ン”という音，のこぎり波なら“ビャ〜ン”という明るい音となります。

FM音源ではこういった波形を作り出すことができるので，PSGとは違った多彩な音が出せるというわけです。

FM音源の仕組み

話は少々難しくなりますが，ここでFM音源の仕組みについて説明しておきましょう。周波数変調については完全には理解しなくてもけっこうですが，オペレータセルとレジスタに関しては知っておいたほうがよいと思います。

周波数変調

昔からあるアナログシンセというのは前に示したいろいろな波形をシンセ本体に持っていて，ある一定以上または以下の周波数しか通さないというフィルタを使ってある波形に含まれている周波数成分を削ることで波形を変えて音を作っていました。しかし，FM音源ではこれ以外の方式を使って波形作り（音色作り）をしています。

FM音源のFMとはFrequency modulationのことで周波数変調という意味です。

周波数変調というのは周波数を次々と変化させることで，たとえばオシレータ（発振器）のつまみを手でくりくり変化させてやるようなものです。といっても実際に手で周波数を変えてやるわけではなく，モジュレータ（変調器）と言われるものでサイン波的に基本の周波数を変化させています。FM音源では基本周波数とモジュレータの周波数が非常に近い（2倍や3倍でも近いという）状態で周波数変調を行いますので，サイン波がゆがんで出てきます。サイン波がゆがむということはすなわち別の波形になってしまったということですから，これをスピーカにつないで音を鳴らすと音色が変化することになります。

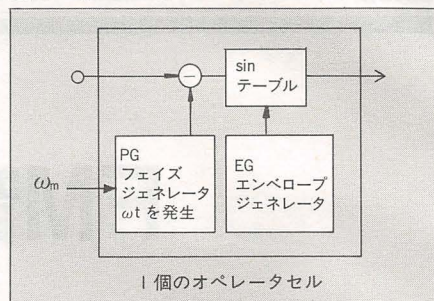
これがFM音源での音色変化の中心的方法になっています。これを式に表すと

$$F=A\sin(\theta+L\sin M\theta)$$

となります。Aは波形の振幅を表すもので音の大きさです。$\theta=\omega t$でωは角速度，tは時間ですから，θは単位時間当たりの角度の変化量を示します。Lはモジュレータのサイン波の振幅で変調の深さ，つまりどれだけ基本の周波数をいじくるかということです。そしてMは基本波形に対するモジュレーションの角速度の変化ということになります。

これを目で見るためのプログラムをBASICで組んでみたのがサンプルのプログラムです。まず試しにL＝1，N＝2としてみてください。すると多少デコボコがあるにしてもなんとなくPSGの波形に近いものができます。これにいろいろ値を入れてやると元のサイン波がどれだけ変化するものかがわかると思います。

OPNではこの基本的な変調を何回か繰り返したり，自分自身の波形で変調をかけてやったりしてさまざまな波形を作ることができます。つまり，これらを上手に組み合わせて波形を作れば，ピアノの波形に近い音なども作れるということになります。

図4　オペレータ

オペレータセル

OPNにはFM変調をかけるやり方などを変えて目的の音を出しやすいように，発振器とエンベロープジェネレータをひとつにまとめたオペレータセル（図4）といわれるものが1音につき4つ，3音なので計12個あります。音色作りには1音内で4つのオペレータセルを使い，これらを上手に組み合わせて使うためのコネクトと呼ばれる組み合わせパターンが8種類あります（図5）。たとえば目的の音がかなり微妙に変化する独特な音なら変調数の多いNo.0や1，2などを使ったり，音色はそんなに微妙でないがそれぞれ周波数の少々違った音を出してウナリを出したいときなどにはNo.5，6，7などを使用します。

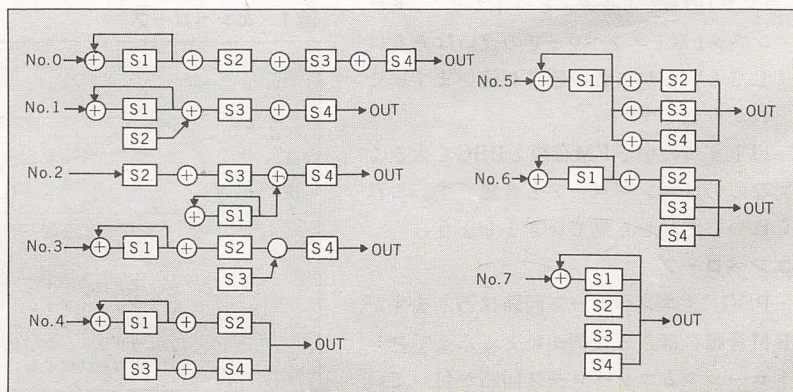
OPNのレジスタ

さて，実際に音色を設定するときはOPNの各レジスタに値を書き込むことで行います。それぞれのレジスタの働きを説明しているとそれだけで何ページにもなってしまいますので，ここでは簡単にまとめたものを表1に示しておくだけにします。シンセサイザをかじったことのある人なら，これだけでだいたいのことはわかると思います。

この中でエンベロープ関係のものは図1を見ればわかるでしょう。そのほかでは，Detuneは各オペレータの出力周波数に微妙

サンプル　FM変調の波形（MZ-1500/2000/2200, X1）

```
100 REM FM ONGEN TEST
101 REM  BY MASAKAZU.T
102 REM
110 INIT"CRT:G"
120 CLS 3
130 LINE[1]0,0,0,199:LINE[1]0,100,319,100
140 Y1=100
150 A=80
160 PRINT "F=A.SIN(T+L.SIN(N.T))"
170 INPUT "L=";L
180 INPUT "N=";N
190 FOR X=1 TO 319 STEP 2
200 Y=100-A*SIN(X*3.14/90+L*SIN(N*X*3.14/90))
210 LINE[7]X-1,Y1,X,Y:Y1=Y
220 NEXT
230 GET A$:IF A$=""THEN 230 ELSE 120
```

図5　オペレータの組み合わせ（コネクト）

なズレを生じさせるためのもので，Multipleは各オペレータが入力周波数の何倍の周波数を出力するかを決めるものです。

さらに詳しく知りたい方は，ヤマハのOPNマニュアルか，MZ-2500，PC-88SR，FMのFM音源カードのマニュアルを見せてもらうかなどしてください。いずれにしても，感覚で理解するしかなさそうですが。

サウンドエディタ

いよいよ今回のメイン，FM音源サウンドエディタの紹介です。これは，FM音源の音色作成と同時に，パソコンを単音シンセサイザとしてこれだけでも十分楽しめるものです。これで作った音色データはあとで発表するミュージックエディタ，ミュージックプレイヤーで活用できるようにしますので，使ってみたい音をたくさん作っておいてくださいね。

プログラムはBASIC版とマシン語版で，マシン語版はS-OSを利用していますので全機種共通です。BASIC版はマシン語版に比べて少し機能が落ちています。

マシン語版はサウンドエディタ（リスト3-A）と音色データ(リスト2)をそれぞれ入力してセーブしてください。音色データは私が参考用に付けたものですから，不要な方は入力しなくてもけっこうです。サウンドエディタとデータ（なくてもよい）をロード後，S-OSからA000H番地にジャンプすれば，サウンドエディタがスタートします。

BASICはプログラム(リスト1)と必要があれば音色データ(リスト2)を入力します。音色データは9900H番地からではなくC800H番地から入力してください。

マシン語版とBASIC版のデータ形式は同じですから，このようにアドレスを変更するだけでデータは共有できるようになっています。

使い方もほとんど同じですので，一緒に説明したいと思います。

使用方法

プログラムをスタートすると，画面に音色No.0のデータが表示され，カーソルマーク［　］が出てエディットモードに入ります。ここでキーボードを使って音を出したり，カーソルを移動させて音作りをすることができます。現在S-OSではカーソルキーがサポートされていませんので，カーソル移動はメインキーで行います。キー操作は図6にまとめておきます。音色データは38個まで扱うことができます。

BASIC版の場合，1度音を出すとスペースキーを押すまで止まりません。マシン語版ではNキーを押すことによりこれをコントロールすることができます。モードの表示はオクターブの横の記号で，キーボードを押しているときだけ音が出る「．」(ピリオド)，1度押すとスペースを押すまで出たままになる「−」，OPNの各パートを順番に使う「＋」，OPN＃0だけ使う「0」が切り替わります。

エディットモードでリターンキーを押すとメニューが出ます。カーソルを上下させてリターンキーを押せば，それぞれのメニューを実行することができます(表2)。この中でBASIC版でサポートしているのはNEIROとNAMEだけです。BASIC版でデータのSAVE，LOADをしたい場合は，プログラムを中断して，モニタなどで各自行ってください。データのアドレスはC800H～CDEFHです。

エディットモードで表示されるデータの内容は，表1のレジスタの内容の省略型になっていますので，慣れるまではそちらを見ながらやってください。

LFOについて

OPNにはLFO(低周波発振器：ビブラートなどに使う）がないため，マシン語版ではタイマAを使用して割り込みを発生させ，

表1　OPNレジスタ

アドレス(16進)	アドレス＋0(チャンネル1)		アドレス＋1(チャンネル2)		アドレス＋2(チャンネル3)		オペレータNo.
21	TEST						
24	TIMER A						
25	TIMER A						
26	TIMER B						
27	CONTROL						
28	KEY ON/OFF						
2D	プリスケーラSET						
2E	〃						
2F	〃						
30 34 38 3C	Detune 7で最大ズレ	Multiple 15で最大倍率	Detune	Multiple	Detune	Multiple	0 2 1 3
40 44 48 4C	Total Level 0で最大音量		Total Level		Total Level		0 2 1 3
50 54 58 5C	Key Scale	Attack Rate 31で最速	Key Scale	Attack Rate	Key Scale	Attack Rate	0 2 1 3
60 64 68 6C		Decay Rate 31で最速		Decay Rate		Decay Rate	0 2 1 3
70 74 78 7C		Sustain Rate 31で最速		Sustain Rate		Sustain Rate	0 2 1 3
80 84 88 8C	Sustain Level 0で最大音	Release Rate 31で最速	Sustain Level	Release Rate	Sustain Level	Release Rate	0 2 1 3
90 94 98 9C		SSG Type Envelop		SSG Type Envelop		SSG Type Envelop	0 2 1 3
A0	F-Numbers1		F-Numbers1		F-Numbers1		
A4	BLOCK	F-N2	BLOCK	F-N2	BLOCK	F-N2	
A8 A9 AA					注アドレス＋0 ．F-Numbers1		3 1 2
AC AD AE					注アドレス＋0 CH3 BLOCK	CH3 F-N2	3 1 2

図6　エディットモードのキー操作

1 −#ラ　3 #ド　4 #レ　6 #ファ　7 #ソ　8 #ラ　0 +#ド

Q −シ　W ド　E レ　R ミ　T ファ　Y ソ　U ラ　I シ　O +ド　P +レ

A +1　S +1　D +5　F +1　K ↑　CR メニュー

Z −1　X −1　C −5　V −1　N モード　M ←　, ↓　. →

オクターブ（A・Z）　データ（S・X）　No.（D・F・C・V）　カーソル（K・M・,・.）

SPACE 音ストップ

表2　メニュー

NEIRO	データメニューにジャンプします。カーソルを移動してリターンキーを押し，音を選びます。
NAME	現在エディット中の音に名前を付けたり，変更したりします。名前は10文字までです。
LOAD	音色データをロードします。
SAVE	ファイルネームを入力してリターンキーを押すと，音色データをセーブします。
COPY	音色データをコピーします。転送データ，転送先の順に選んでリターンキーを押します。
EXIT	S-OSに戻ります。

ソフト的にLFOを作っています。LFOでの変調は三角波によるもののみで，かけられるものは基本周波のトータルレベル（音の大きさ）と周波数のみです。パラメータは4つあり，第1が周波数にかけるか音量にかけるかのFN/TL切り替えで，0で周波数，1で音量となります。第2は音の鳴り始めからどれだけ時間がたってからLFOを作動させるかというS.WAITで，0で最小，255で最長となります。第3はDEEPで，どれだけの深さの変調をかけてやるかをこれで決めます。変化量は0～125までで，0でLFO OFFとなります。第4はSTEPで，変調の深さを何回の割り込みで1ずつ深くしていくかというものです。つまりLFOの周波数はDEEPとこのSTEPによって決まります。

単音シンセサイザですがこのままでも十分楽しめるものと思います。できるだけいいアンプと大きなスピーカにつないでFMサウンドを味わってください。なるべく近いうちにミュージックエディタとミュージックプレイヤーを発表する予定です。そのための音色データもしっかり貯えておきましょう。

リスト1　サウンドエディタ(BASIC版)

```
10 REM ******************
11 REM  ﾈｲﾛ ｻﾞｸﾘ ﾌﾟﾛｸﾞﾗﾑ
12 REM
13 REM    BY M.TAGA
14 REM ******************
15 REM
20 CLS:CLR:RESTORE
30 DIM D(3,10),F(1,16),K(90)
40 REM KEY CODE READ---------------------------------------
50 FOR I=0 TO 90:K(I)=255:NEXT
60 FOR I=0 TO 29
70 READ A:K(A)=I
71 K(A)=I
80 NEXT
90 REM ONKAI DATA READ-------------------------------------
100 FOR I=0 TO 16
110 READ F(0,I),F(1,I)
120 NEXT
130 NO=0:DA=$C800
200 REM ==========
201 REM  CRT MAKE
202 REM ==========
203 REM
210 CLS
220 CURSOR 2,1:PRINT "NO.";NO
230 PRINT TAB(12);"1     2     3     4"
240 PRINT " CONNECT          ---   ---   ---"
250 PRINT " F.BACK           ---   ---   ---"
260 PRINT " MULIPLE"
270 PRINT " T.LEVEL"
280 PRINT " A.RATE"
290 PRINT " D.RATE"
300 PRINT " S.RATE"
310 PRINT " S.LEVEL"
320 PRINT " R.RATE"
330 PRINT " DETUNE"
340 PRINT " K.SCALE"
345 CURSOR 1,20:PRINT"OCT"
350 A=DA+NO*40
360 CURSOR 15,1:FOR I=0 TO 9
362 B=PEEK(A):IF B<$20 THEN B=$20
364 PRINT CHR$(B);:A=A+1:NEXT:A=A+1
370 FOR I=0 TO 3:B=PEEK(A):A=A+1:Q=INT(B/8):R=B-Q*8:D(I,2)
    =R:D(I,9)=Q:NEXT


    4)=R:D(I,10)=Q:NEXT
400 FOR I=0 TO 3:B=PEEK(A):A=A+1:D(I,5)=B:NEXT
410 FOR I=0 TO 3:B=PEEK(A):A=A+1:D(I,6)=B:NEXT
420 FOR I=0 TO 3:B=PEEK(A):A=A+1:Q=INT(B/16):R=B-Q*16:D(I,
    8)=R:D(I,7)=Q:NEXT
430 B=PEEK(A):Q=INT(B/8):R=B-Q*8:D(0,0)=R:D(0,1)=Q
440 FOR I=1 TO 10:B=D(1,I):D(1,I)=D(2,I):D(2,I)=B:NEXT
450 X=0:Y=0:GOSUB 5000:GOSUB 5400
460 X=0:Y=1:GOSUB 5000:GOSUB 5400
470 FOR X=0 TO 3:FOR Y=2 TO 10:GOSUB 5000:GOSUB 5400:NEXT:
    NEXT
480 X=0:Y=0:O=4:CURSOR 7,20:PRINT O
490 GOSUB 5200
500 REM ===========
501 REM  KEY INPUT
502 REM ===========
503 REM
510 GET A$:A=ASC(A$)
520 IF A>90 THEN 510
530 A=K(A)
540 IF A<17 THEN 600
550 ON A-16 GOTO 700,750,810,820,830,840,500,1100,1600,120
    0,1300,1400,1500
560 GOTO 510
600 REM ===========
601 REM  ONKAI KEY
602 REM ===========
603 REM
610 OUT@0,$A4:OUT@1,F(0,A)+8*O
620 OUT@0,$A0:OUT@1,F(1,A)
630 OUT@0,$28:OUT@1,$0:OUT@1,$F0
640 GOTO 500
700 REM ========
701 REM  OCT UP
702 REM ========
703 REM
710 O=O+1:IF O=8 THEN O=0
720 CURSOR 7,20:PRINT O
730 GOTO 500
750 REM ==========
751 REM  OCT DOWN
752 REM ==========
753 REM
760 O=O-1:IF O=-1 THEN O=7
770 GOTO 720
800 REM ==============
801 REM  DATA UP DOWN
802 REM ==============
803 REM
810 D=1:GOTO 850
820 D=-1:GOTO 850
830 D=5:GOTO 850
840 D=-5:GOTO 850
850 ON Y+1 GOTO 860,860,870,880,890,890,890,870,870,860,900
860 LM=7:GOTO 910
870 LM=15:GOTO 910
880 LM=127:GOTO 910
890 LM=31:GOTO 910
900 LM=3:GOTO 910
910 D(X,Y)=D(X,Y)+D
920 IF D(X,Y)<0 THEN D(X,Y)=D(X,Y)+LM+1
930 IF D(X,Y)>LM THEN D(X,Y)=D(X,Y)-LM-1
940 GOSUB 5000:GOSUB 5400
950 GOTO 500
1100 REM ==========
1101 REM  OTO STOP
1102 REM ==========
1103 REM
1110 OUT@0,$28:OUT@1,0
1120 GOTO 500
```

```
1200 REM ===========
1201 REM  CURSOR UP
1202 REM ===========
1203 REM
1210 GOSUB 5300
1220 Y=Y-1
1230 IF (Y=-1)+(Y=1)*(X>0) THEN Y=10
1240 GOSUB 5200
1250 GOTO 500
1300 REM =============
1301 REM  CURSOR DOWN
1302 REM =============
1303 REM
1310 GOSUB 5300
1320 Y=Y+1
1330 IF (X=0)*(Y=11) THEN Y=0:GOTO 1350
1340 IF Y=11 THEN Y=2
1350 GOSUB 5200
1360 GOTO 500
1400 REM =============
1401 REM  CURSOR LEFT
1402 REM =============
1403 REM
1410 IF Y<2 THEN 1460
1420 GOSUB 5300
1430 X=X-1
1440 IF X=-1 THEN X=3
1450 GOSUB 5200
1460 GOTO 500
1500 REM ==============
1501 REM  CURSOR RIGHT
1502 REM ==============
1503 REM
1510 IF Y<2 THEN 1560
1520 GOSUB 5300
1530 X=X+1
1540 IF X=4 THEN X=0
1550 GOSUB 5200
1560 GOTO 500
1600 REM ======
1601 REM  MENW
1602 REM ======
1603 REM
1610 GOSUB 5300
1620 CURSOR 20,16:PRINT"1:NEIRO"
1630 CURSOR 20,17:PRINT"2:NAME"
1640 GET A:IF A=0 THEN 1640
1650 IF A=1 THEN 1900
1660 IF A=2 THEN 1800
1670 CURSOR 20,16:PRINT"       "
1680 CURSOR 20,17:PRINT"      "
1690 GOSUB 5200
1700 GOTO 500
1800 REM ======
1801 REM  NAME
1802 REM ======
1803 REM
1810 CURSOR 20,16:PRINT"       "
1820 CURSOR 20,19:INPUT A$
1830 A=DA+NO*40
1840 FOR I=1 TO 10:POKE A,ASC(MID$(A$+"          ",I,1))
     :A=A+1:NEXT
1850 CURSOR 15,1:PRINT LEFT$(A$+"          ",10)
1860 CURSOR 0,16:FOR I=16 TO 23:CURSOR 20,I:PRINT SPC(19
     ):NEXT
1870 GOSUB 5200
1880 GOTO 500
1900 REM =======
1901 REM  NEIRO
1902 REM =======
1903 REM
1910 CLS
1920 Y=2:PP=11:GOSUB 5500:REM MULTIPLE
1930 Y=3:PP=15:GOSUB 5500:REM T.LEVEL
1940 Y=4:PP=19:GOSUB 5500:REM A.RATE
1950 Y=5:PP=23:GOSUB 5500:REM D.R.
1960 Y=6:PP=27:GOSUB 5500:REM S.R.
1965 Y=7:PP=31:GOSUB 5500:REM S.L.
1970 X=0:Y=0:GOSUB 5000:POKE DA+NO*40+35,OD
2020 CURSOR 16,1:PRINT"======="
2030 CURSOR 16,2:PRINT" NEIRO "
2040 CURSOR 16,3:PRINT"======="
2050 PRINT"    NO.  NAME        NO.  NAME"
2060 A=DA
2070 FOR J=0 TO 1
2080 FOR I=0 TO 18
2090 CURSOR 5+J*17,5+I
2110 PRINT RIGHT$(" "+STR$(I+J*19),2);":";
2120 FOR II=0 TO 9
2122 JJ=PEEK(A):IF JJ<$20 THEN JJ=$20
2124 PRINT CHR$(JJ);
2126 A=A+1:NEXT
2130 A=A+30
2140 NEXT:NEXT
2150 CURSOR 10,24:INPUT"NO.?";A
2160 A=INT(ABS(A)):IF A>37 THEN 2050
2170 NO=A
2180 FOR I=0 TO 18
2190 GOTO 200
5000 REM ====================
5001 REM  DATA WRITE
5002 REM   X,Y;DAT  NO SET
5003 REM   OD ;OPN WRITE DATA
5004 REM ====================
5005 REM
5010 ON Y+1 GOTO 5020,5020,5030,5040,5050,5040,5040,5060,50
     60,5030,5050
5015 STOP
5020 P=$B0:OD=D(0,0)+D(0,1)*8:GOTO 5070
5030 P=$30:OD=D(X,0)+D(X,9)*8:GOTO 5070
5040 P=$10*(Y+1):OD=D(X,Y):GOTO 5070
5050 P=$50:OD=D(X,4)+D(X,10)*64:GOTO 5070
5060 P=$80:OD=D(X,7)*16+D(X,8):GOTO 5070
5070 ON X+1 GOTO 5080,5090,5100,5110
5075 STOP
5080 PX=0:GOTO 5120
5090 PX=8:GOTO 5120
5100 PX=4:GOTO 5120
5110 PX=12:GOTO 5120
5120 OUT@0,P+PX:OUT@1,OD
5130 RETURN
5200 REM =================
5201 REM  CURSOR SET
5202 REM   X,Y;CURSOR ADR
5203 REM =================
5204 REM
5210 CURSOR 10+X*5,3+Y:PRINT"["
5220 CURSOR 14+X*5,3+Y:PRINT"]"
5230 RETURN
5300 REM =================
5301 REM  CURSOR RESET
5302 REM   X,Y;CURSOR ADR
5303 REM =================
5304 REM
5310 CURSOR 10+X*5,3+Y:PRINT" "
5320 CURSOR 14+X*5,3+Y:PRINT" "
5330 RETURN
5400 REM ================
5401 REM  DATA PRINT
5402 REM   X,Y;CURSOR
5404 REM ================
5405 REM
5410 CURSOR 11+X*5,3+Y
5420 PRINT RIGHT$("  "+STR$(D(X,Y)),3)
5430 RETURN
5500 REM =========
5501 REM  KAKIKOMI
5502 REM =========
5503 REM
5510 X=0
5520 AA=DA+NO*40
5530 FOR I=0 TO 3
5540 GOSUB 5000
5550 X=X+2
5560 IF X=4 THEN X=1
5570 POKE AA+PP+I,OD
5580 NEXT
5590 RETURN
10000 REM ===============
10001 REM  KEY CODE DATA
10002 REM ===============
10003 REM
10010 DATA 49,81,87,51,69,52,82,84,54,89
10020 DATA 55,85,56,73,79,48,80,65,90,83
10030 DATA 88,68,67,78,32,13,75,44,77,46
10050 REM =================
10051 REM  C,D,E,.. NO DATA
10052 REM =================
10053 REM
10060 DATA 2,38,2,71,2,105,2,142,2,181,2,222
10070 DATA 3,10,3,56,3,105,3,157,3,212
10080 DATA 4,14,4,76,4,141,4,211
10090 DATA 5,28,5,106
```

●SP-5030用変更点

```
610 OUT#0,$A4:OUT#1,F(0,A)+8*O
620 OUT#0,$A0:OUT#1,F(1,A)
630 OUT#0,$28:OUT#1,$0:OUT@1,$F0
1110 OUT#0,$28:OUT#1,0
5120 OUT#0,P+PX:OUT#1,OD
```

●HuBASIC用変更点

```
130 NO=0:DA=&HC800
362 B=PEEK(A):IF B<&H20 THEN B=&H20
610 OUT (0),&HA4:OUT (1),F(0,A)+8*O
620 OUT (0),&HA0:OUT (1),F(1,A)
630 OUT (0),&H28:OUT (1),&H0:OUT (1),&HF0
1110 OUT (0),&H28:OUT (1),0
2122 JJ=PEEK(A):IF JJ<&H20 THEN JJ=&H20
5020 P=&HB0:OD=D(0,0)+D(0,1)*8:GOTO 5070
5030 P=&H30:OD=D(X,0)+D(X,9)*8:GOTO 5070
5040 P=&H10*(Y+1):OD=D(X,Y):GOTO 5070
5050 P=&H50:OD=D(X,4)+D(X,10)*64:GOTO 5070
5060 P=&H80:OD=D(X,7)*16+D(X,8):GOTO 5070
5120 OUT (0),P+PX:OUT (1),OD
```

リスト 2　音色データ(BASIC版で使うときはC800H番地から入力)

```
9900 42 52 41 53 53 20 31 20 :EC
9908 0D 0D 0D 01 23 01 11 1B :78
9910 2B 22 12 8D 4F 15 52 06 :A8
9918 08 0E 03 02 00 00 00 18 :33
9920 18 28 28 3A 00 00 00 00 :A2
9928 42 52 41 53 53 20 32 20 :ED
9930 0D 0D 0D 01 27 01 11 1A :7B
9938 22 36 10 8E 59 4F 52 0A :FA
9940 0D 0B 03 00 03 00 00 15 :33
9948 FF 25 28 3A 00 00 00 00 :86
9950 54 52 55 4D 50 45 54 20 :51
9958 0D 0D 0D 01 07 01 01 0D :3E
9960 1E 1C 10 8E 8E 8D 53 0E :54
9968 0E 0E 03 00 00 00 00 13 :32
9970 FA 13 0A 3A 00 00 00 00 :51
9978 53 54 52 49 4E 47 20 31 :28
--------------------------------
SUM: F1 6C E5 98 CE C0 F1 31 :8A

9980 20 0D 0D 11 15 01 11 1C :8E
9988 12 31 11 59 59 5C 4E 0A :BA
9990 0B 0D 04 00 00 00 00 15 :31
9998 58 26 06 3A 00 00 00 00 :BE
99A0 53 54 52 49 4E 47 20 32 :29
99A8 20 0D 0D 11 15 01 11 1D :8F
99B0 22 3A 10 5C 53 5C 4D 07 :CB
99B8 09 07 04 00 00 00 00 13 :27
99C0 34 13 07 3A 00 00 00 00 :88
99C8 45 50 49 41 4E 4F 20 31 :0D
99D0 20 0D 0D 11 21 0A 11 1A :A1
99D8 12 3C 13 9A 98 DA D8 0F :54
99E0 0C 07 0C 00 05 03 05 26 :52
99E8 28 46 28 14 00 00 00 00 :AA
99F0 45 50 49 41 4E 4F 20 32 :0E
99F8 20 0D 0D 3F 41 01 01 2C :E8
--------------------------------
SUM: 77 69 95 14 BF 87 0C 82 :5D

9A00 1D 2C 10 9F DB 9E 5E 0F :DE
9A08 07 06 07 08 0A 0B 00 8A :BB
9A10 86 F6 F7 1C 00 00 00 00 :8F
9A18 45 50 49 41 4E 4F 20 33 :0F
9A20 20 0D 0D 31 4C 64 01 25 :41
9A28 43 33 10 9C DB 9C DA 04 :77
9A30 09 04 03 03 01 03 00 17 :2E
9A38 02 06 A5 3A 00 00 00 00 :E7
9A40 47 55 49 54 41 52 20 0D :F9
9A48 0D 0D 0D 32 33 33 31 2A :1A
9A50 28 2E 10 DF DF DF 9F 07 :A9
9A58 04 04 0A 04 04 04 03 F7 :18
9A60 FB FB 0B 21 00 00 00 00 :22
9A68 45 42 41 53 53 20 31 20 :DF
9A70 0D 0D 0D 36 35 30 31 1E :11
9A78 3A 16 10 DF DF 9F 9F 07 :63
--------------------------------
SUM: 64 B6 F5 00 19 52 4D 86 :4D

9A80 06 09 06 07 06 06 08 29 :59
9A88 19 19 F9 20 00 00 00 00 :4B
9A90 45 42 41 53 53 20 32 20 :E0
9A98 0D 0D 0D 30 30 30 30 17 :FE
9AA0 3F 17 10 9E D8 DC DC 0E :A2
9AA8 0A 04 05 08 08 08 08 B6 :E9
9AB0 B6 B6 B6 30 00 00 00 00 :52
9AB8 45 4F 52 47 41 4E 20 31 :0D
9AC0 20 0D 0D 35 58 72 31 27 :91
9AC8 1E 29 18 DF 1F 1F 1F 12 :AD
9AD0 04 0F 0F 00 00 00 00 2F :51
9AD8 0F 0F 0F 3D 00 00 00 00 :6A
9AE0 45 4F 52 47 41 4E 20 32 :0E
9AE8 20 0D 0D 66 31 73 22 1C :82
9AF0 15 15 1B 1C 98 1F 1F 12 :49
9AF8 0F 0F 0F 00 00 00 00 FF :2C
--------------------------------
SUM: 8F 65 36 E1 2B F9 1F 1C :6A
```

```
9B00 0F 0F 0F 1F 00 00 00 00 :4C
9B08 50 4F 52 47 41 4E 20 31 :18
9B10 20 0D 0D 78 74 60 10 20 :B6
9B18 25 26 14 94 90 92 92 02 :A9
9B20 02 02 02 00 00 00 00 07 :0D
9B28 07 05 06 04 00 00 00 00 :16
9B30 50 4F 52 47 41 4E 20 32 :19
9B38 20 0D 0D 14 12 0C 06 26 :98
9B40 20 24 1F 92 12 5E 59 0A :C8
9B48 04 06 05 02 00 02 00 47 :5A
9B50 08 19 09 1C 00 00 00 00 :46
9B58 46 4C 55 54 45 20 0D 0D :BA
9B60 0D 0D 0D 72 02 31 01 2A :F7
9B68 35 27 10 DF 54 D0 8F 09 :07
9B70 07 0B 04 03 00 00 00 E5 :FE
9B78 F5 25 08 3B 00 00 00 00 :5D
--------------------------------
SUM: CD E7 94 64 45 1B DE 28 :12

9B80 50 49 43 43 4F 4C 4F 20 :29
9B88 0D 0D 0D 01 01 01 01 2E :59
9B90 1D 1E 1C 1F 14 14 14 0A :BC
9B98 06 0B 0B 0F 00 00 00 59 :84
9BA0 F9 19 19 3E 00 00 00 00 :69
9BA8 4F 42 4F 45 20 0D 0D 0D :6C
9BB0 0D 0D 0D 31 39 32 34 20 :17
9BB8 2B 33 13 D9 DC 56 54 0B :DB
9BC0 0C 00 06 00 0C 00 00 13 :31
9BC8 5B 1B 0B 3A 00 00 00 00 :BB
9BD0 43 4C 41 52 49 4E 45 54 :52
9BD8 20 0D 0D 01 02 04 01 27 :69
9BE0 25 37 13 5F 13 5F 53 00 :93
9BE8 0B 0F 0F 00 00 00 00 00 :29
9BF0 4B 09 09 3B 00 00 00 00 :98
9BF8 47 4C 4F 43 4B 45 4E 20 :23
--------------------------------
SUM: 8C 29 D8 69 4E EC E0 97 :A7

9C00 0D 0D 0D 3B 01 07 71 23 :FE
9C08 1C 18 10 DF DE 5F DE 0E :4C
9C10 09 10 07 00 07 05 04 FF :2F
9C18 16 A0 17 1C 00 00 00 00 :E9
9C20 56 49 42 52 50 48 4E 20 :39
9C28 0D 0D 0D 39 56 05 01 22 :DE
9C30 1F 46 11 5F DB 9E 9E 10 :FC
9C38 07 0C 05 00 0A 0B 0A BA :F1
9C40 71 F6 F5 24 00 00 00 00 :80
9C48 58 59 4C 4F 50 48 4E 20 :52
9C50 0D 0D 0D 35 21 33 31 21 :02
9C58 1C 23 11 59 59 59 59 17 :CB
9C60 0E 14 0E 0A 0B 0B 0C EB :44
9C68 F8 E8 F8 3C 00 00 00 00 :14
9C70 4B 4F 54 4F 20 0D 0D 0D :84
9C78 0D 0D 0D 33 34 31 31 20 :10
--------------------------------
SUM: 21 54 66 E9 9A 7E 6C A9 :F1

9C80 20 1F 16 DA DD DC DF 08 :CF
9C88 05 04 0A 05 04 02 03 27 :48
9C90 14 36 15 38 00 00 00 00 :97
9C98 5A 49 54 41 52 20 0D 0D :C4
9CA0 0D 0D 0D 37 39 21 31 21 :0A
9CA8 4A 22 17 9F 9F 9F 9F 06 :05
9CB0 06 0C 0C 06 06 06 06 01 :37
9CB8 01 01 05 02 00 00 00 00 :09
9CC0 43 4C 41 56 20 0D 0D 0D :6D
9CC8 0D 0D 0D 3C 3B 30 31 2D :2C
9CD0 24 13 13 DF 1F 1F DF 04 :4A
9CD8 05 04 01 04 04 04 02 F7 :0F
9CE0 07 17 AC 3B 00 00 00 00 :05
9CE8 48 41 52 50 53 49 43 20 :2A
9CF0 0D 0D 0D 0C 1F 41 55 28 :10
9CF8 3C 20 16 1F DF 1F 9F 0C :3A
--------------------------------
SUM: 02 D3 41 61 E0 CD 1B ED :2C
```

```
9D00 02 0C 05 04 04 04 07 1A :40
9D08 F6 06 27 3A 00 00 00 00 :5D
9D10 42 45 4C 4C 20 0D 0D 0D :66
9D18 0D 0D 0D 32 20 32 32 21 :FE
9D20 24 2B 1C 1D 1D CA 5E 04 :D1
9D28 04 04 04 01 03 04 03 20 :37
9D30 03 30 03 1C 00 00 00 00 :52
9D38 48 41 52 50 20 0D 0D 0D :72
9D40 0D 0D 0D 02 01 02 01 23 :50
9D48 28 35 19 5F 5F 1F 9F 0C :FE
9D50 0D 06 0C 00 06 05 07 F4 :25
9D58 15 15 14 39 00 00 00 00 :77
9D60 48 41 52 4D 4F 4E 49 20 :2E
9D68 0D 0D 0D 35 05 10 01 26 :98
9D70 33 20 15 14 14 10 0E 05 :B3
9D78 08 02 08 00 00 00 00 99 :AB
--------------------------------
SUM: A1 D1 BC 76 52 B2 B3 80 :DB

9D80 09 09 19 38 00 00 00 00 :63
9D88 54 49 4D 50 41 4E 49 20 :32
9D90 0D 0D 0D 70 50 70 00 28 :7F
9D98 28 27 10 1D 1D 1D 1F 06 :DB
9DA0 00 0D 00 00 06 06 07 F4 :14
9DA8 04 24 05 39 00 00 00 00 :66
9DB0 4D 41 52 53 20 54 4F 20 :16
9DB8 0D 0D 0D 38 31 32 32 0F :03
9DC0 2C 06 06 9F 1C 9F 53 0D :F2
9DC8 08 04 09 10 09 08 00 22 :58
9DD0 E2 D2 F2 39 00 00 00 00 :DF
9DD8 57 49 4E 44 20 42 45 4C :25
9DE0 4C 53 0D 36 3D 21 35 18 :8D
9DE8 14 06 07 1C 5C 4E 4D 08 :3C
9DF0 07 04 09 00 00 00 00 13 :27
9DF8 10 52 32 04 00 00 00 00 :98
--------------------------------
SUM: D4 D9 85 5B E3 BF 0A 1F :58

9E00 53 50 41 43 45 20 54 41 :21
9E08 4C 4B 0D 33 32 33 31 06 :73
9E10 06 06 06 05 06 03 08 1F :47
9E18 1F 1F 1F 01 03 04 02 01 :68
9E20 01 01 09 05 00 00 00 00 :10
9E28 52 4E 44 28 31 29 20 0D :93
9E30 0D 0D 0D 50 01 0B 50 0B :DE
9E38 02 0E 05 58 17 46 42 0F :1B
9E40 02 01 00 0B 08 18 17 52 :97
9E48 C2 42 D2 05 00 00 00 00 :DB
9E50 42 52 45 41 4B 20 47 4C :18
9E58 41 53 0D 3D 38 3F 3B 06 :96
9E60 26 06 0F 5F 1F 1F 1F 0A :01
9E68 1F 05 05 00 00 03 03 A4 :D3
9E70 06 74 74 3C 00 00 00 00 :2A
9E78 4F 53 48 41 4D 49 20 0D :EE
--------------------------------
SUM: 07 E4 C6 BB C0 B6 1C ED :EB

9E80 0D 0D 0D 51 61 69 03 0E :53
9E88 2E 2B 06 1E 9F DF 9F 15 :AF
9E90 02 1B 0D 00 03 08 0B F3 :33
9E98 03 31 63 2A 00 00 00 00 :C1
9EA0 0D 0D 0D 0D 0D 0D 0D 0D :68
9EA8 0D 0D 0D 00 00 00 00 00 :27
9EB0 00 00 00 00 00 00 00 00 :00
9EB8 00 00 00 00 00 00 00 00 :00
9EC0 00 00 00 00 00 00 00 00 :00
9EC8 0D 0D 0D 0D 0D 0D 0D 0D :68
9ED0 0D 0D 0D 00 00 00 00 00 :27
9ED8 00 00 00 00 00 00 00 00 :00
9EE0 00 00 00 00 00 00 00 00 :00
9EE8 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: 74 B8 B7 B3 1D 6A C7 30 :14
```

リスト3-A　サウンドエディタ(S-OS版)ダンプリスト

```
A000 C3 32 A0 C3 41 A0 CA AB :AE
A008 C3 00 00 C3 F4 1F C3 F4 :50
A010 AD C3 E8 1F C3 EE 1F C3 :0A
A018 F7 1F C3 F1 1F C3 D3 1F :9E
A020 C3 AF 1F C3 AC 1F C3 A9 :8B
```

```
A028 1F C3 A6 1F C3 A3 1F C3 :EF
A030 9D 1F 21 00 99 36 00 11 :BD
A038 00 99 13 01 F0 05 0B ED :9A
A040 B0 CD 7B AB AF 32 67 AC :97
A048 21 0A 99 11 28 00 06 26 :29
```

```
A050 3E 0D 77 19 10 FC 21 78 :80
A058 1F 5E 23 56 D5 DD E1 CD :56
A060 17 A0 7C FE 00 CA 81 A0 :1C
A068 FE 01 CA 99 A0 FE 02 CA :CC
A070 BC A0 FE 10 CA A1 A0 FE :73
```


▶『それ行け! X1』の読者の方々, Vol.7のp.26の上の方に載っている名前の10番目をよーく見てください。それがぼくです。モニタになっちゃった。それだけのことでした。
森本　雄一郎 (16) 大阪府

```
A078 11 CA C2 A0 FE 20 CA C8 :ED
--------------------------------
SUM: B9 8B F8 EB 33 01 C8 32 :55

A080 A0 00 00 00 3E B0 32 07 :C7
A088 E0 3E C3 32 38 10 21 CA :46
A090 AB 22 39 10 ED 56 C3 EC :08
A098 A0 00 00 00 3E B0 32 07 :C7
A0A0 E0 3E A0 ED 47 01 7D 00 :70
A0A8 3E 4F ED 79 00 3E 06 ED :24
A0B0 79 00 3E F3 ED 79 00 ED :FD
A0B8 5E C3 EC A0 00 00 00 C3 :70
A0C0 99 A0 00 00 00 C3 A1 A0 :3D
A0C8 00 00 00 3E 00 ED 47 01 :73
A0D0 7D 00 3E 4F ED 79 00 3E :AE
A0D8 54 ED 79 00 3E F3 ED 79 :51
A0E0 00 21 CA AB 22 54 00 ED :F9
A0E8 5E C3 EC A0 FB 3E 0C CD :BF
A0F0 0B A0 DD 36 00 02 DD 36 :D3
A0F8 01 01 11 DF AC CD 11 A0 :1C
--------------------------------
SUM: 94 C2 0E 28 C9 FB 9A 49 :33

A100 CD 14 A0 11 E2 AC CD 11 :FE
A108 A0 CD 14 A0 11 FF AC CD :AA
A110 11 A0 CD 14 A0 11 1D AD :0D
A118 CD 11 A0 CD 14 A0 11 3B :4B
A120 AD CD 11 A0 CD 14 A0 11 :BD
A128 45 AD CD 11 A0 CD 14 A0 :F1
A130 11 4E AD CD 11 A0 CD 14 :6B
A138 A0 11 56 AD CD 11 A0 CD :FF
A140 14 A0 11 5E AD CD 11 A0 :4E
A148 CD 14 A0 11 66 AD CD 11 :83
A150 A0 CD 14 A0 11 6F AD CD :1B
A158 11 A0 CD 14 A0 11 77 AD :67
A160 CD 11 A0 CD 14 A0 11 7F :8F
A168 AD CD 11 A0 CD 14 A0 11 :BD
A170 88 AD CD 11 A0 CD 14 A0 :34
A178 11 8C AD CD 11 A0 CD 14 :A9
--------------------------------
SUM: 93 A3 BF 2B 48 09 5C C7 :94

A180 A0 11 93 AD CD 11 A0 CD :3C
A188 14 A0 11 9B AD CD 11 A0 :8B
A190 CD 14 A0 11 A1 AD CD 11 :BE
A198 A0 CD 14 A0 CD 14 A0 11 :B3
A1A0 A7 AD CD 11 A0 3A 67 AC :1F
A1A8 26 00 6F 29 29 29 E5 29 :1E
A1B0 29 D1 19 11 00 99 19 22 :F8
A1B8 61 AC 3E 04 32 65 AC DD :6F
A1C0 36 00 07 DD 36 01 14 3E :A3
A1C8 34 CD 0B A0 DD 36 00 09 :C8
A1D0 DD 36 01 14 3E 2E CD 0B :6C
A1D8 A0 3E 30 CD 0B A0 AF 32 :67
A1E0 5E AC 32 5F AC 32 63 AC :88
A1E8 32 66 AC F5 CD 21 AA 00 :D1
A1F0 47 CD 19 A9 F1 3C 32 5F :94
A1F8 AC FE 0F 20 EE 06 01 78 :46
--------------------------------
SUM: E2 DA 34 C3 97 9A FF 6A :4D

A200 32 5E AC 0E 02 79 32 5F :56
A208 AC C5 CD 21 AA 00 47 CD :1D
A210 19 A9 C1 0C 3E 0B B9 20 :B1
A218 EC 04 3E 04 B8 20 E0 3A :24
A220 67 AC DD 36 00 04 DD 36 :3D
A228 01 01 CD DD A8 3E 0F DD :7E
A230 77 00 3E 01 DD 77 01 ED :F8
A238 5B 61 AC CD 11 A0 AF 32 :C7
A240 5E AC 32 5F AC CD 94 A8 :50
A248 CD 0E A0 FE 00 28 F9 32 :CC
A250 64 AC 21 BC AC 01 20 00 :BA
A258 ED B9 20 EC 79 FE 11 DA :14
A260 05 A3 D6 11 16 00 5F 87 :8B
A268 83 5F 21 6F A2 19 E9 C3 :D9
A270 0A A4 C3 06 A4 C3 2B A4 :AD
A278 C3 27 A4 C3 33 A4 C3 2F :1A
--------------------------------
SUM: EE CA 7D 6E 98 71 A2 89 :D7

A280 A4 C3 D2 A7 C3 47 A8 C3 :55
A288 F8 A7 C3 1D A8 C3 7A A4 :08
A290 C3 D3 A2 C3 9C A2 C3 5B :57
A298 A4 C3 6A A4 DD 36 00 09 :91
A2A0 DD 36 01 14 3A 63 AC 3C :AD
A2A8 FE 04 20 01 AF 32 63 AC :13
A2B0 E6 01 28 04 3E 2D 18 02 :98
A2B8 3E 2E CD 0B A0 3A 63 AC :2D
A2C0 E6 02 28 04 3E 2B 18 02 :97
A2C8 3E 30 CD 0B A0 CD 71 A8 :CC
A2D0 C3 48 A2 16 00 F3 01 00 :B7
A2D8 00 3E 28 ED 79 00 03 7A :49
A2E0 00 00 ED 79 00 01 02 00 :69
A2E8 3E 28 ED 79 00 03 7A 00 :49
A2F0 00 ED 79 00 14 3E 03 00 :BB
A2F8 00 00 BA 20 D9 FB AF 32 :8F
--------------------------------
SUM: 27 36 83 73 EF 06 2A B7 :29

A300 66 AC C3 48 A2 87 16 00 :5C
A308 5F 21 BD AC 19 5E 23 56 :D9
A310 3A 65 AC 87 87 87 B2 57 :E9
A318 26 00 3A 66 AC 87 6F 01 :69
A320 07 AC 09 73 23 72 3A 66 :64
A328 AC CB 47 28 05 01 02 00 :EE
A330 18 03 01 00 00 CB 3F C6 :EC
A338 A4 F3 ED 79 00 03 7A 00 :7A
A340 00 ED 79 00 0B 3A 66 AC :BD
A348 CB 3F C6 A0 00 00 00 00 :70
A350 00 00 00 00 00 00 00 00 :00
A358 00 00 00 00 00 00 ED 79 :66
A360 00 03 7B 00 00 ED 79 00 :E4
A368 3A 66 AC 16 00 5F AF 21 :91
A370 13 AC 19 77 21 25 AC 19 :5A
A378 77 3C 21 1F AC 19 77 2A :59
--------------------------------
SUM: 23 1C 44 41 EE F8 ED 63 :FA

A380 61 AC 11 25 00 19 C5 46 :67
A388 3A 66 AC 16 00 5F 21 19 :FB
A390 AC 19 70 C1 0B 3E 28 ED :54
A398 79 00 03 3A 63 AC E6 01 :AC
A3A0 28 2B 3A 66 AC CB 3F ED :96
A3A8 79 00 3A 66 AC CB 3F C6 :95
A3B0 F0 00 00 00 00 00 00 00 :F0
A3B8 00 00 00 00 00 00 00 00 :00
A3C0 00 00 00 00 ED 79 00 FB :61
A3C8 CD 7F A8 18 22 3A 66 AC :7A
A3D0 CB 3F C6 F0 ED 79 00 FB :21
A3D8 CD 7F A8 F3 0B 3E 28 ED :45
A3E0 79 00 00 00 00 03 3A 66 :1C
A3E8 AC CB 3F ED 79 00 FB 3A :51
A3F0 63 AC E6 02 CA 48 A2 3A :E5
A3F8 66 AC 3C FE 06 20 01 AF :22
--------------------------------
SUM: A4 B6 1B EA 16 CD D8 18 :32

A400 32 66 AC C3 48 A2 3E FF :2E
A408 18 02 3E 01 21 65 AC 86 :11
A410 E6 07 77 DD 36 00 07 DD :5B
A418 36 01 14 3E 30 86 CD 0B :17
A420 A0 CD 71 A8 C3 48 A2 06 :39
A428 FF 18 0A 06 01 18 06 06 :4C
A430 FB 18 02 06 05 21 70 AC :5D
A438 3A 5F AC 16 00 5F 87 83 :C4
A440 5F 19 7E CB 3F 38 04 CB :07
A448 20 18 F8 C5 CD 21 AA C1 :4E
A450 80 47 CD 19 A9 CD 8B A8 :56
A458 C3 48 A2 3A 67 AC 3C FE :34
A460 26 20 01 AF 32 67 AC C3 :FE
A468 EC A0 3A 67 AC 3D FE FF :13
A470 20 02 3E 25 32 67 AC C3 :8D
A478 EC A0 CD AA A8 DD 36 00 :BE
--------------------------------
SUM: 1A EE C9 71 6C 27 5E 5F :92

A480 16 DD 36 01 10 11 B1 AD :A9
A488 CD 11 A0 DD 36 00 16 DD :84
A490 36 01 11 11 AC AD CD 11 :90
A498 A0 DD 36 00 16 DD 36 01 :DD
A4A0 12 11 B7 AD CD 11 A0 DD :E2
A4A8 36 00 16 DD 36 01 13 11 :84
A4B0 BC AD CD 11 A0 DD 36 00 :FA
A4B8 16 DD 36 01 14 11 C1 AD :BD
A4C0 CD 11 A0 DD 36 00 16 DD :84
A4C8 36 01 15 11 C6 AD CD 11 :AE
A4D0 A0 06 10 CD 71 A8 CD 8B :F4
A4D8 A8 DD 36 00 14 DD 70 01 :1D
A4E0 3E 3E CD 0B A0 CD 0E A0 :6F
A4E8 FE 4B 28 0A FE 2C 28 1C :E9
A4F0 FE 0D 28 2E 18 E0 DD 36 :6C
A4F8 00 14 DD 70 01 3E 20 CD :8D
--------------------------------
SUM: 58 06 E2 F9 F7 E4 C7 70 :4B

A500 0B A0 05 3E 0F B8 20 CE :A3
A508 06 14 18 CA DD 36 00 14 :23
A510 DD 70 01 3E 20 CD 0B A0 :24
A518 04 3E 16 B8 20 B8 06 10 :FE
A520 18 B4 78 FE 11 CA 4B A5 :0D
A528 FE 10 CA 98 A5 FE 12 CA :EF
A530 79 A7 FE 13 CA 17 A7 FE :B7
A538 14 CA AE A6 C3 08 A0 CD :6A
A540 97 A7 CD 71 A8 CD 94 A8 :2D
A548 C3 48 A2 DD 36 00 15 DD :B2
A550 36 01 15 11 AC AD CD 11 :94
A558 A0 CD 71 A8 DD 36 01 16 :B0
A560 CD AD A7 DA 3F A5 D5 DD :91
A568 36 00 0F DD 36 01 01 11 :6B
A570 D3 AD CD 11 A0 D1 DD 36 :E2
A578 00 0F DD 36 01 01 2A 61 :AF
--------------------------------
SUM: 9B BD 77 52 EC 82 29 FD :B5

A580 AC 06 0A AF B9 3E 20 28 :AA
A588 01 1A 77 CD 0B A0 0D 23 :3A
A590 13 10 F0 36 0D C3 3F A5 :FD
A598 CD A5 A5 CD 49 A6 78 32 :7D
A5A0 67 AC C3 EC A0 3E 0C CD :79
A5A8 0B A0 DD 36 00 10 DD 36 :E1
A5B0 01 01 11 EC AD CD 11 A0 :2A
A5B8 DD 36 00 11 DD 36 01 02 :3A
A5C0 11 B1 AD CD 11 A0 DD 36 :00
A5C8 00 10 DD 36 01 03 11 EC :24
A5D0 AD CD 11 A0 DD 36 00 05 :43
A5D8 DD 36 01 04 11 DF AC CD :81
A5E0 11 A0 DD 36 00 0A 11 AC :8B
A5E8 AD CD 11 A0 DD 36 00 16 :54
A5F0 11 DF AC CD 11 A0 DD 36 :2D
A5F8 00 1B 11 AC AD CD 11 A0 :03
--------------------------------
SUM: 47 83 0E 94 DF FD 78 53 :13

A600 06 00 DD 36 01 05 11 00 :30
A608 99 21 F8 02 19 EB DD 36 :CB
A610 00 04 78 CD DD A8 3E 3A :46
A618 CD 0B A0 EB CD 11 A0 DD :BE
A620 36 00 15 3E 13 80 CD DD :C6
A628 A8 3E 3A CD 0B A0 EB CD :50
A630 11 A0 D5 11 28 00 19 D1 :A9
A638 E5 21 28 00 19 EB E1 04 :17
A640 CD 14 A0 3E 13 B8 20 C6 :70
A648 C9 3A 67 AC 47 CD 71 A8 :43
A650 CD 6B A6 3E 3E CD 0B A0 :D2
A658 CD 8B A8 CD 0E A0 FE 4B :C4
A660 28 24 FE 2C 28 34 FE 0D :DD
A668 C8 18 F0 3E 12 B8 38 0B :1B
A670 3E 05 80 DD 36 00 04 DD :B7
A678 77 01 C9 3E F2 80 DD 36 :04
--------------------------------
SUM: 15 B5 C5 86 2B 12 2F 50 :D1

A680 00 15 DD 77 01 C9 CD 6B :6B
A688 A6 3E 20 CD 0B A0 05 3E :BF
A690 FF B8 C2 50 A6 06 25 C3 :5D
A698 50 A6 CD 6B A6 3E 20 CD :FF
A6A0 0B A0 04 3E 26 B8 C2 50 :DD
A6A8 A6 06 00 C3 50 A6 CD A5 :D7
A6B0 A5 DD 36 00 05 DD 36 01 :D1
A6B8 00 11 C6 AD CD 11 A0 DD :DF
A6C0 36 00 07 DD 36 01 01 11 :63
A6C8 CB AD CD 11 A0 CD 49 A6 :B2
A6D0 78 32 67 AC DD 36 00 04 :D4
A6D8 DD 36 01 01 CD DD A8 CD :34
A6E0 49 A6 3A 67 AC 4F 78 32 :35
A6E8 67 AC 79 6F 26 00 29 29 :73
A6F0 29 E5 29 29 D1 19 11 00 :5B
A6F8 99 19 E5 3A 67 AC 6F 26 :79
--------------------------------
SUM: 13 AA 89 81 2A EE 8F 15 :83

A700 00 29 29 29 E5 29 29 D1 :83
A708 19 11 00 99 19 D1 EB 01 :99
A710 28 00 ED B0 C3 EC A0 CD :E1
A718 97 A7 DD 36 00 15 DD 36 :79
A720 01 10 11 E7 AD CD 11 A0 :34
A728 CD 1A A0 11 AC AD CD 11 :CF
A730 A0 CD 71 A8 DD 36 01 11 :AB
A738 CD AD A7 DA 3F A5 3E 01 :1E
A740 CD 2C A0 FD 2A 74 1F FD :50
A748 36 11 0D 21 00 99 22 70 :A0
```

▶ウーン，困った。どーしよー。年末にアルバイトしたお金で，さて何を買おうか。候補その1，MZ-1P17を買う。その2，I/Oボックスを買ってFM音源ボードを製作する。その3，I/Oボックス＋漢ROM＋New BASICを買う。その4，安いCDを買う。と，「とらぬたぬきの皮算用」をしている私でありました。誰か何を買えばいいのか教えてください。
高杉　淳（15）東京都

```
A750 1F 21 F0 05 22 72 1F DD :C5
A758 36 00 15 DD 36 01 14 CD :40
A760 20 A0 DA 3F A5 CD 23 A0 :0E
A768 DD 36 00 00 DD 36 01 15 :3C
A770 11 D3 AD CD 11 A0 C3 3F :11
A778 A5 CD 97 A7 DD 36 00 15 :D8
--------------------------------
SUM: 1E 59 8C D5 28 A9 09 B8 :6A

A780 DD 36 01 10 CD 26 A0 DA :91
A788 3F A5 DD 36 00 15 CD 2F :08
A790 A0 CD 29 A0 C3 98 A5 06 :3C
A798 08 DD 36 01 10 DD 36 00 :3F
A7A0 14 11 D3 AD CD 11 A0 CD :F0
A7A8 14 A0 10 F1 C9 DD 36 00 :91
A7B0 15 21 76 1F 5E 23 56 CD :6F
A7B8 1D A0 1A FE 1B 20 02 37 :49
A7C0 C9 21 15 00 19 E5 AF 4F :FB
A7C8 BE 28 04 23 0C 18 F9 D1 :FB
A7D0 B7 C9 3A 5F AC FE 02 DA :9F
A7D8 48 A2 FE 0B D2 48 A2 3A :E9
A7E0 5E AC 3D FE FF 20 02 3E :A4
A7E8 03 CD AA A8 32 5E AC CD :2B
A7F0 94 A8 CD 8B A8 C3 48 A2 :E9
A7F8 3A 5F AC FE 02 DA 48 A2 :09
--------------------------------
SUM: D3 2B 61 5E 2D 3F 00 63 :8C

A800 FE 0B D2 48 A2 3A 5E AC :09
A808 3C FE 04 20 01 AF CD AA :85
A810 A8 32 5E AC CD 94 A8 CD :BA
A818 8B A8 C3 48 A2 CD AA A8 :FF
A820 3A 5F AC 3D FE FF 20 04 :A3
A828 3E 0E 18 0F FE 01 20 0B :9D
A830 47 3A 5E AC FE 01 78 38 :3A
A838 02 3E 0A 32 5F AC CD 94 :E8
A840 A8 CD 8B A8 C3 48 A2 CD :22
A848 AA A8 3A 5F AC 3C FE 0F :E0
A850 20 03 AF 18 10 FE 0B 20 :23
A858 0C 47 3A 5E AC FE 01 78 :0E
A860 DA 65 A8 3E 02 32 5F AC :64
A868 CD 94 A8 CD 8B A8 C3 48 :14
A870 A2 C5 06 03 CD 0E A0 FE :E9
A878 00 20 F7 10 F7 C1 C9 E5 :8D
--------------------------------
SUM: F5 65 1E 21 E7 20 39 F1 :CA

A880 21 64 AC CD 0E A0 BE 28 :92
A888 FA E1 C9 21 70 17 2B 7C :F3
A890 B5 20 FB C9 F5 C5 CD C0 :E0
A898 A8 3E 5B CD 0B A0 78 DD :0E
A8A0 77 00 3E 5D CD 0B A0 C1 :4B
A8A8 F1 C9 F5 C5 CD C0 A8 3E :E7
A8B0 20 CD 0B A0 78 DD 77 00 :64
A8B8 3E 20 CD 0B A0 C1 F1 C9 :51
A8C0 3A 5E AC 47 87 87 80 C6 :DF
A8C8 0A DD 77 00 C6 04 47 3A :A9
A8D0 5F AC C6 03 FE 0E 38 01 :19
A8D8 3C DD 77 01 C9 C5 D5 16 :0A
A8E0 00 0E 30 FE 64 38 05 0C :E9
A8E8 D6 64 18 F7 CD 02 A9 FE :BF
A8F0 0A 38 05 0C D6 0A 18 F7 :42
A8F8 CD 02 A9 81 CD 0B A0 D1 :42
--------------------------------
SUM: CA C9 2C 1E 18 32 18 F2 :31

A900 C1 C9 47 AF BA 79 20 0A :DD
A908 FE 30 20 04 3E 20 18 02 :CA
A910 16 01 CD 0B A0 78 0E 30 :45
A918 C9 3A 5E AC 5F 87 87 83 :FD
A920 C6 0B DD 77 00 3A 5F AC :6A
A928 C6 03 FE 0E 38 01 3C DD :27
A930 77 01 21 70 AC 3A 5F AC :FA
A938 16 00 5F 87 83 5F 19 7E :75
A940 4F A0 47 C5 CB 39 38 04 :3B
A948 CB 3F 18 F8 C1 CD DD A8 :2D
A950 23 5E E5 2A 61 AC 19 3A :F0
A958 5E AC CB 3F 30 02 CB CF :E0
A960 32 60 AC 5F 19 79 2F A6 :04
A968 B0 77 E1 23 5F 56 3E 03 :21
A970 BA 38 4F 28 0F 3D BA 28 :97
A978 12 3D BA 28 45 7B 21 4F :61
--------------------------------
SUM: 00 78 92 DE 47 A7 21 47 :3E

A980 AC C3 BB A9 7B 21 31 AC :4C
A988 C3 BB A9 7B 21 2B AC CD :67
```

```
A990 BB A9 2A 5E AC E5 2A 5F :06
A998 AC E5 3E 03 32 5F AC AF :BE
A9A0 32 5E AC F5 CD 21 AA 47 :10
A9A8 CD 19 A9 F1 3C 06 04 BF :85
A9B0 20 EE E1 22 5F AC E1 22 :1F
A9B8 5E AC C9 06 06 77 23 10 :89
A9C0 FC C9 3A 60 AC 87 87 82 :9B
A9C8 57 26 03 F3 01 00 00 7A :EE
A9D0 ED 79 00 03 7B 00 00 ED :D1
A9D8 79 00 01 02 00 7A ED 79 :5C
A9E0 00 03 7B 00 00 ED 79 00 :E4
A9E8 14 00 00 00 25 20 DD FB :31
A9F0 3A 5F AC FE 03 28 05 FE :71
A9F8 00 28 15 C9 3A 60 AC 4F :9B
--------------------------------
SUM: 5A 0F 45 B2 72 70 E0 69 :8B

AA00 06 00 21 37 AC 09 06 06 :1F
AA08 73 23 23 23 23 10 F9 C9 :D1
AA10 21 68 AC 16 00 3E 07 A3 :33
AA18 5F 19 7E 21 55 AC C3 BB :96
AA20 A9 21 70 AC 3A 5F AC 16 :41
AA28 00 5F 87 83 5F 19 46 23 :4A
AA30 5E 2A 61 AC 19 3A 5E AC :F2
AA38 CB 3F 30 02 CB CF 5F 19 :4E
AA40 78 A6 C9 08 D9 3E 01 32 :39
AA48 5C AC 01 02 00 3E 27 00 :70
AA50 00 00 00 00 00 ED 79 00 :66
AA58 03 3A 06 AC 00 ED 79 00 :55
AA60 0E 00 06 00 21 2B AC 09 :15
AA68 56 AF BE CA 60 AB 21 19 :D2
AA70 AC 09 BE 28 04 35 C3 60 :F7
AA78 AB 21 25 AC 09 BE 28 04 :90
--------------------------------
SUM: 5D F2 6D C2 08 A3 4A E3 :56

AA80 35 C3 60 AB 21 31 AC 09 :0A
AA88 7E 21 25 AC 09 77 21 13 :24
AA90 AC 09 7E BA 28 05 82 28 :C4
AA98 02 18 07 21 1F AC 09 7E :94
AAA0 2F 77 21 1F AC 09 7E 21 :3A
AAA8 13 AC 09 86 77 5F 32 5B :B1
AAB0 AC 21 4F AC 09 AF BE CA :08
AAB8 07 AB C5 21 55 AC 09 EB :8D
AAC0 CB 21 CB 21 21 37 AC 09 :E5
AAC8 CB 39 CB 39 CB 39 38 08 :4C
AAD0 3E 40 81 01 00 00 18 06 :1E
AAD8 3E 40 81 01 02 00 F5 1A :11
AAE0 57 F1 1E 04 CB 3A 30 15 :B4
AAE8 00 00 00 00 00 00 00 ED :ED
AAF0 79 00 0C F5 3A 5B AC 86 :41
AAF8 ED 79 00 0D F1 23 C6 04 :51
--------------------------------
SUM: 25 38 0A 06 D6 44 62 B0 :99

AB00 1D 20 E1 C1 C3 60 AB C5 :72
AB08 CB 21 21 07 AC 09 CB 39 :CD
AB10 7E 23 56 83 38 07 CB 7B :FF
AB18 28 08 15 18 05 CB 7B 20 :C8
AB20 01 14 5F 79 CB 3F C6 A4 :61
AB28 CB 41 28 05 01 02 00 18 :54
AB30 03 01 00 00 ED 79 00 0C :76
AB38 00 00 00 00 ED 51 00 0D :4B
AB40 D6 04 00 00 00 00 00 00 :DA
AB48 00 00 00 00 00 00 00 00 :00
AB50 00 00 00 00 ED 79 00 0C :72
AB58 00 00 00 00 ED 59 00 C1 :07
AB60 0C 3E 06 B9 C2 62 AA AF :86
AB68 32 5C AC 3A 5D AC FE 00 :7B
AB70 20 05 08 D9 FB ED 4D 08 :43
AB78 D9 FB C9 01 02 00 3E 25 :03
--------------------------------
SUM: 6A 60 77 AE 48 13 B5 17 :16

AB80 ED 79 00 03 AF 00 00 ED :05
AB88 79 00 0B 3E 24 00 00 00 :E6
AB90 00 00 00 00 00 00 00 00 :00
AB98 00 00 00 00 00 00 00 ED :ED
ABA0 79 00 03 3E 6E 00 ED 79 :8E
ABA8 00 0B 3E 27 00 00 00 00 :70
ABB0 00 00 00 00 00 00 00 00 :00
ABB8 00 00 00 00 00 00 ED 79 :66
ABC0 00 03 3A 06 AC 00 ED 79 :55
ABC8 00 C9 C5 F5 01 02 00 3E :C4
ABD0 27 00 00 00 00 00 00 00 :27
ABD8 00 ED 79 00 03 3A 06 AC :55
ABE0 ED 79 00 0B 00 00 00 00 :71
```

```
ABE8 00 00 00 00 00 00 00 00 :00
ABF0 00 00 00 00 00 00 00 00 :00
ABF8 ED 78 20 05 F1 C1 C3 43 :42
--------------------------------
SUM: E0 2E E4 B1 E2 FD 90 72 :84

AC00 AA F1 C1 C3 43 AA 35 00 :41
AC08 00 00 00 00 00 00 00 00 :00
AC10 00 00 00 00 00 00 00 00 :00
AC18 00 00 00 00 00 00 00 00 :00
AC20 00 00 00 00 00 00 00 00 :00
AC28 00 00 00 00 00 00 00 00 :00
AC30 00 00 00 00 00 00 00 00 :00
AC38 00 00 00 00 00 00 00 00 :00
AC40 00 00 00 00 00 00 00 00 :00
AC48 00 00 00 00 00 00 00 00 :00
AC50 00 00 00 00 00 00 00 00 :00
AC58 00 00 00 00 00 00 00 00 :00
AC60 00 00 00 00 00 00 00 00 :00
AC68 08 08 08 08 0C 0E 0E 0F :57
AC70 07 23 B0 38 23 B0 0F 0B :FF
AC78 30 7F 0F 40 1F 13 50 1F :9F
--------------------------------
SUM: E9 9B 88 43 91 7B A2 39 :36

AC80 17 60 1F 1B 70 F0 1F 80 :B0
AC88 0F 1F 80 70 0B 30 C0 13 :2C
AC90 50 01 24 00 FF 25 01 7F :19
AC98 26 02 FF 27 03 31 51 57 :2A
ACA0 33 45 34 52 54 36 59 37 :18
ACA8 55 38 49 4F 30 50 41 5A :40
ACB0 53 58 44 43 4D 2C 2E 4B :24
ACB8 0D 20 4E 46 56 26 02 47 :86
ACC0 02 69 02 8E 02 B5 02 DE :92
ACC8 02 0A 03 38 03 69 03 9D :53
ACD0 03 D4 03 0E 04 4C 04 8D :C9
ACD8 04 D3 04 1C 05 6A 05 4E :B9
ACE0 4F 0D 20 20 20 20 20 20 :1C
ACE8 20 20 20 20 20 20 31 20 :11
ACF0 20 20 20 32 20 20 20 20 :12
ACF8 33 20 20 20 20 34 0D 20 :14
--------------------------------
SUM: 51 FE 5D 5E 32 B6 87 62 :DB

AD00 43 4F 4E 4E 45 43 54 20 :2A
AD08 20 20 20 20 20 20 20 2D :0D
AD10 2D 2D 20 20 2D 2D 2D 20 :41
AD18 20 2D 2D 2D 0D 20 46 2E :48
AD20 42 41 43 4B 20 20 20 20 :91
AD28 20 20 20 20 20 2D 2D 2D :27
AD30 20 20 2D 2D 2D 20 20 2D :34
AD38 2D 2D 0D 20 4D 55 4C 54 :C9
AD40 49 50 4C 45 0D 20 54 2E :D9
AD48 4C 45 56 45 4C 0D 20 41 :E6
AD50 2E 52 41 54 45 0D 20 44 :CB
AD58 2E 52 41 54 45 0D 20 53 :DA
AD60 2E 52 41 54 45 0D 20 53 :DA
AD68 2E 4C 45 56 45 4C 0D 20 :D3
AD70 52 2E 52 41 54 45 0D 20 :D9
AD78 44 45 54 55 4E 45 0D 20 :F2
--------------------------------
SUM: 42 C1 A8 E5 68 9C 9B 22 :51

AD80 4B 2E 53 43 41 4C 45 0D :EE
AD88 4C 46 4F 0D 20 46 4E 2F :D1
AD90 54 4C 0D 20 53 2E 57 41 :E6
AD98 49 54 0D 20 44 45 45 50 :E8
ADA0 0D 20 53 54 45 50 0D 20 :96
ADA8 4F 43 54 0D 4E 41 4D 45 :14
ADB0 0D 4E 45 49 52 4F 0D 4C :E3
ADB8 4F 41 44 0D 53 41 56 45 :10
ADC0 0D 43 4F 50 59 0D 45 58 :F2
ADC8 49 54 0D 3E 3D 0D 20 20 :72
ADD0 20 20 20 20 20 20 20 20 :00
ADD8 20 20 20 20 20 20 20 20 :00
ADE0 20 20 20 20 20 20 0D 46 :13
ADE8 49 4C 45 0D 3D 3D 3D 3D :DB
ADF0 3D 3D 3D 0D FB C3 D0 1F :71
--------------------------------
SUM: 28 86 2A 4F 5E A0 AB 1D :ED
```


▶10月のある日，友人のO.Kに12月号p.180の川田君の意見と同じことを，ぼくのMZ-2000でやってみせた（ちなみにぼくは以前からこのことを知っていたのです）。そのあと彼が，自分のMZをいじっていると，テープが出てこないことに気づいた。彼はぼくがやってみせたと同じようにして直してみたが，電源を入れてもテープが動いてくれない。彼のマシンはMZ-2200である。MZ-1T02を分解して原因を調べてみたが異常はない。もとに戻そ

リスト3-B サウンドエディタソースリスト

```
0001: 0000                      ;
0002: 0000                      ; OTOZUKURI
0003: 0000                      ;    BY M.TAGA
0004: 0000                      ;
0005: 0000                      ; EQU TBL
0006: 0000                      ;
0007: 0000 (0000)               OPN#0:  EQU     0               ;OPN#0 I/O POTE
0008: 0000 (0002)               OPN#1:  EQU     2               ;OPN#1 I/O POTE
0009: 0000 (0040)               TLSADR: EQU     40H             ;TOTAL LEVEL SUB ADR
0010: 0000 (0004)               SADRSD: EQU     4               ;SUB ADR SRIDE DATA
0011: 0000 (000F)               IO#B:   EQU     0FH             ;I/O B SUB ADR
0012: 0000 (00A4)               FNHSAD: EQU     A4H             ;F.NO HIGH SUB ADR
0013: 0000 (00A0)               FNLSAD: EQU     A0H             ;F.NO LOW SUB ADR
0014: 0000 (0028)               KESADR: EQU     28H             ;KEY ON/OFF SUB ADR
0015: 0000 (05F0)               ODATAS: EQU     1520            ;O DATA SIZE
0016: 0000 (1F78)               #XYADR: EQU     1F78H           ;CURSOR ADR
0017: 0000 (1F76)               #KBFAD: EQU     1F76H           ;KEY B ADR(GETL)
0018: 0000 (1F74)               #IBFAD: EQU     1F74H           ;INF BLOCK
0019: 0000 (1F72)               #SIZE:  EQU     1F72H           ;FILE SIZE
0020: 0000 (1F70)               #DTADR: EQU     1F70H           ;FILE TOP ADR
0021: 0000 (0000)               #VER0:  EQU     00H             ;MZ-80K
0022: 0000 (0001)               #VER1:  EQU     01H             ;MZ-700
0023: 0000 (0002)               #VER2:  EQU     02H             ;MZ-1500
0024: 0000 (0010)               #VER4:  EQU     10H             ;MZ-80B
0025: 0000 (0011)               #VER5:  EQU     11H             ;MZ-2000
0026: 0000 (0020)               #VER8:  EQU     20H             ;X1
0027: 0000 (000C)               CHRCLS: EQU     0CH             ;CRT CLEAR
0028: 0000 (9900)               ODATA:  EQU     9900H           ;NEIRO DATA TOP ADR.
0029: 0000 (990A)               ODATA2: EQU     990AH           ;NEIRO DATA 2
0030: 0000 (E007)               CP8253: EQU     E007H           ;8253 CONT ADR.
0031: 0000 (00B0)               PITM0:  EQU     B0H             ;8253,C2,M0
0032: 0000 (00A0)               IREG:   EQU     A0H             ;M2 INT ADR HIGH
0033: 0000 (0006)               VREG:   EQU     06H             ;M2 INT ADR LOW
0034: 0000 (0000)               IREGX1: EQU     00H             ;X1 M2 INT ADR HIGH
0035: 0000 (0054)               VREGX1: EQU     54H             ;X1 M2 INT ADR LOW
0036: 0000 (0026)               DATASU: EQU     38              ;DATA SUU
0037: 0000 (0028)               DATALO: EQU     0040            ;DATA LONG
0038: 0000 (000D)               CHRCR:  EQU     0DH             ;CR
0039: 0000 (00C3)               COMJP:  EQU     C3H             ;JP COMMAND
0040: 0000 (1038)               MOJPL:  EQU     1038H           ;MO JP ADR(JP)
0041: 0000 (1039)               MOJPH:  EQU     1039H           ;MO JP ADR2(ADR.)
0042: 0000 (007C)               PIODAT: EQU     007CH           ;PIO DATA I/O
0043: 0000 (007D)               PIOCOM: EQU     007DH           ;PIO COMMAND I/O
0044: 0000 (004F)               PIOMD1: EQU     4FH             ;PIO M1 CONTROLE COMMAND
0045: 0000 (00F3)               PIOIE:  EQU     F3H             ;PIO INT E
0046: 0000 (002E)               CHRTEN: EQU     2EH             ;.
0047: 0000 (0030)               CHRZER: EQU     30H             ;0
0048: 0000 (0020)               KEYSUU: EQU     32              ;USR KEY SUU
0049: 0000 (0011)               NOTEKY: EQU     17              ;NOTE KEY SUU
0050: 0000 (002D)               CHRMIN: EQU     2DH             ;-
0051: 0000 (002B)               CHRPUR: EQU     2BH             ;+
0052: 0000 (0025)               SWTOF:  EQU     37              ;STEP WAIT OFFSET
0053: 0000 (00F0)               SALLON: EQU     F0H             ;SLOT ALL ON
0054: 0000 (02F8)               ODMID:  EQU     760             ;OTO DATA MID OFFSET
0055: 0000 (003E)               CHRDAI: EQU     3EH             ;>
0056: 0000 (004B)               CHRK:   EQU     4BH             ;K
0057: 0000 (002C)               CHRCON: EQU     2CH             ;,
0058: 0000 (0020)               CHRSPC: EQU     20H             ;SPACE
0059: 0000 (003A)               CHRCOR: EQU     3AH             ;COLON
0060: 0000 (0001)               OBJINF: EQU     1               ;INF OBJ
0061: 0000 (001B)               CHRBRK: EQU     1BH             ;SIFT+BREAK
0062: 0000 (005B)               CHRDCO: EQU     5BH             ;[
0063: 0000 (005D)               CHRDCC: EQU     5DH             ;]
0064: 0000 (1770)               KEYWSU: EQU     6000            ;KEY WAIT SUU
0065: 0000 (0054)               X1JPTB: EQU     0054H           ;X1 JP TBL
0066: 0000                      ;
0067: A000                              ORG     A000H
0068: A000                      ;
0069: A000 C332A0                       JP      COLDHA
0070: A003 C341A0                       JP      HAJIME
0071: A006                      ;
0072: A006                      ; INT JP TBL
0073: A006                      ;
0074: A006 CAAB                         DEFW    INTOPN
0075: A008                      ;
0076: A008                      ; JP TBL
0077: A008                      ;
0078: A008 C30000               EXIT:   JP      0000H
0079: A00B C3F41F               #PRINT: JP      1FF4H
0080: A00E C3F4AD               #GETKY: JP      EIGKEY
0081: A011 C3E81F               #MSG:   JP      1FE8H
0082: A014 C3EE1F               #LTNL:  JP      1FEEH
0083: A017 C3F71F               #VER:   JP      1FF7H
0084: A01A C3F11F               #PRNTS: JP      1FF1H
0085: A01D C3D31F               #GETL:  JP      1FD3H
0086: A020 C3AF1F               #WRI:   JP      1FAFH
0087: A023 C3AC1F               #WRD:   JP      1FACH
0088: A026 C3A91F               #RDI:   JP      1FA9H
0089: A029 C3A61F               #RDD:   JP      1FA6H
0090: A02C C3A31F               #FILE:  JP      1FA3H
0091: A02F C39D1F               #FPRNT: JP      1F9DH
0092: A032                      ;
0093: A032                      ;
0094: A032                      ; START
0095: A032                      ;
0096: A032 210099               COLDHA: LD      HL,ODATA        ;COLD START
0097: A035 3600                         LD      (HL),0
0098: A037 110099                       LD      DE,ODATA
0099: A03A 13                           INC     DE
0100: A03B 01F005                       LD      BC,ODATAS
0101: A03E 0B                           DEC     BC
0102: A03F EDB0                         LDIR
0103: A041 CD7BAB               HAJIME: CALL    LFOSS
0104: A044 AF                           XOR     A
0105: A045 3267AC                       LD      (DATNO),A
0106: A048 210A99                       LD      HL,ODATA2
0107: A04B 112800                       LD      DE,DATALO
0108: A04E 0626                         LD      B,DATASU
0109: A050 3E0D                         LD      A,CHRCR
0110: A052 77                   HAJIM2: LD      (HL),A
0111: A053 19                           ADD     HL,DE
0112: A054 10FC                         DJNZ    HAJIM2
0113: A056 21781F                       LD      HL,#XYADR
0114: A059 5E                           LD      E,(HL)
0115: A05A 23                           INC     HL
0116: A05B 56                           LD      D,(HL)
0117: A05C D5                           PUSH    DE
0118: A05D DDE1                         POP     IX
0119: A05F CD17A0                       CALL    #VER
0120: A062 7C                           LD      A,H
0121: A063 FE00                         CP      #VER0
0122: A065 CA81A0                       JP      Z,MZ80
0123: A068 FE01                         CP      #VER1
0124: A06A CA99A0                       JP      Z,MZ700
0125: A06D FE02                         CP      #VER2
0126: A06F CABCA0                       JP      Z,MZ1500
0127: A072 FE10                         CP      #VER4
0128: A074 CAA1A0                       JP      Z,MZ80B
0129: A077 FE11                         CP      #VER5
0130: A079 CAC2A0                       JP      Z,MZ2000
0131: A07C FE20                         CP      #VER8
0132: A07E CAC8A0                       JP      Z,X1
0133: A081                      ;MZ-80K/C/1200
0134: A081 00                   MZ80:   NOP
0135: A082 00                           NOP
0136: A083 00                           NOP
0137: A084 3EB0                         LD      A,PITM0
0138: A086 3207E0                       LD      (CP8253),A
0139: A089 3EC3                         LD      A,COMJP
0140: A08B 323810                       LD      (MOJPL),A
0141: A08E 21CAAB                       LD      HL,INTOPN
0142: A091 223910                       LD      (MOJPH),HL
0143: A094 ED56                         IM      1
0144: A096 C3ECA0                       JP      CRTMAK
0145: A099                      ;MZ-700
0146: A099 00                   MZ700:  NOP
0147: A09A 00                           NOP
0148: A09B 00                           NOP
0149: A09C 3EB0                         LD      A,PITM0
0150: A09E 3207E0                       LD      (CP8253),A
0151: A0A1 3EA0                 MZ80B:  LD      A,IREG
0152: A0A3 ED47                         LD      I,A
0153: A0A5 017D00                       LD      BC,PIOCOM
0154: A0A8 3E4F                         LD      A,PIOMD1
0155: A0AA ED79                         OUT     (C),A
0156: A0AC 00                           NOP
0157: A0AD 3E06                         LD      A,VREG
0158: A0AF ED79                         OUT     (C),A
0159: A0B1 00                           NOP
0160: A0B2 3EF3                         LD      A,PIOIE
0161: A0B4 ED79                         OUT     (C),A
0162: A0B6 00                           NOP
0163: A0B7 ED5E                         IM      2
0164: A0B9 C3ECA0                       JP      CRTMAK
0165: A0BC                      ;MZ-1500
0166: A0BC 00                   MZ1500: NOP
0167: A0BD 00                           NOP
0168: A0BE 00                           NOP
0169: A0BF C399A0                       JP      MZ700
0170: A0C2                      ;MZ-2000
0171: A0C2 00                   MZ2000: NOP
0172: A0C3 00                           NOP
0173: A0C4 00                           NOP
0174: A0C5 C3A1A0                       JP      MZ80B
0175: A0C8                      ; X1 HENKOU
0176: A0C8 00                   X1:     NOP
0177: A0C9 00                           NOP
0178: A0CA 00                           NOP
0179: A0CB 3E00                         LD      A,IREGX1
0180: A0CD ED47                         LD      I,A
0181: A0CF 017D00                       LD      BC,PIOCOM
0182: A0D2 3E4F                         LD      A,PIOMD1
0183: A0D4 ED79                         OUT     (C),A
0184: A0D6 00                           NOP
0185: A0D7 3E54                         LD      A,VREGX1
0186: A0D9 ED79                         OUT     (C),A
0187: A0DB 00                           NOP
0188: A0DC 3EF3                         LD      A,PIOIE
0189: A0DE ED79                         OUT     (C),A
0190: A0E0 00                           NOP
0191: A0E1 21CAAB                       LD      HL,INTOPN
0192: A0E4 225400                       LD      (X1JPTB),HL
0193: A0E7 ED5E                         IM      2
0194: A0E9 C3ECA0                       JP      CRTMAK
0195: A0EC                      ;
0196: A0EC                      ; CRT MAKE
0197: A0EC FB                   CRTMAK: EI
0198: A0ED 3E0C                         LD      A,CHRCLS        ;CRT CLS
0199: A0EF CD0BA0                       CALL    #PRINT
0200: A0F2 DD360002                     LD      (IX+0),2
0201: A0F6 DD360101                     LD      (IX+1),1
0202: A0FA 11DFAC                       LD      DE,M1
0203: A0FD CD11A0                       CALL    #MSG
0204: A100 CD14A0                       CALL    #LTNL
0205: A103 11E2AC                       LD      DE,M2
0206: A106 CD11A0                       CALL    #MSG
0207: A109 CD14A0                       CALL    #LTNL
0208: A10C 11FFAC                       LD      DE,M3
0209: A10F CD11A0                       CALL    #MSG
0210: A112 CD14A0                       CALL    #LTNL
0211: A115 111DAD                       LD      DE,M4
0212: A118 CD11A0                       CALL    #MSG
0213: A11B CD14A0                       CALL    #LTNL
0214: A11E 113BAD                       LD      DE,M5
0215: A121 CD11A0                       CALL    #MSG
0216: A124 CD14A0                       CALL    #LTNL
0217: A127 1145AD                       LD      DE,M6
0218: A12A CD11A0                       CALL    #MSG
0219: A12D CD14A0                       CALL    #LTNL
0220: A130 114EAD                       LD      DE,M7
0221: A133 CD11A0                       CALL    #MSG
0222: A136 CD14A0                       CALL    #LTNL
0223: A139 1156AD                       LD      DE,M8
0224: A13C CD11A0                       CALL    #MSG
0225: A13F CD14A0                       CALL    #LTNL
0226: A142 115EAD                       LD      DE,M9
0227: A145 CD11A0                       CALL    #MSG
0228: A148 CD14A0                       CALL    #LTNL
0229: A14B 1166AD                       LD      DE,M10
0230: A14E CD11A0                       CALL    #MSG
0231: A151 CD14A0                       CALL    #LTNL
0232: A154 116FAD                       LD      DE,M11
0233: A157 CD11A0                       CALL    #MSG
0234: A15A CD14A0                       CALL    #LTNL
0235: A15D 1177AD                       LD      DE,M12
0236: A160 CD11A0                       CALL    #MSG
0237: A163 CD14A0                       CALL    #LTNL
0238: A166 117FAD                       LD      DE,M13
0239: A169 CD11A0                       CALL    #MSG
0240: A16C CD14A0                       CALL    #LTNL
0241: A16F 1188AD                       LD      DE,M14
0242: A172 CD11A0                       CALL    #MSG
0243: A175 CD14A0                       CALL    #LTNL
0244: A178 118CAD                       LD      DE,M15
0245: A17B CD11A0                       CALL    #MSG
0246: A17E CD14A0                       CALL    #LTNL
0247: A181 1193AD                       LD      DE,M16
0248: A184 CD11A0                       CALL    #MSG
0249: A187 CD14A0                       CALL    #LTNL
0250: A18A 119BAD                       LD      DE,M17
0251: A18D CD11A0                       CALL    #MSG
0252: A190 CD14A0                       CALL    #LTNL
0253: A193 11A1AD                       LD      DE,M18
0254: A196 CD11A0                       CALL    #MSG
0255: A199 CD14A0                       CALL    #LTNL
0256: A19C CD14A0                       CALL    #LTNL
0257: A19F 11A7AD                       LD      DE,M19
0258: A1A2 CD11A0                       CALL    #MSG
0259: A1A5                      ;
0260: A1A5                      ;
0261: A1A5 3A67AC                       LD      A,(DATNO)       ;DATA ADR SET
0262: A1A8 2600                         LD      H,0
0263: A1AA 6F                           LD      L,A
0264: A1AB 29                           ADD     HL,HL
0265: A1AC 29                           ADD     HL,HL
0266: A1AD 29                           ADD     HL,HL
0267: A1AE E5                           PUSH    HL
0268: A1AF 29                           ADD     HL,HL
0269: A1B0 29                           ADD     HL,HL
0270: A1B1 D1                           POP     DE
```

うとするとネジが3本もあまってしまった。もう一度分解すると今度はハンダ付けされていたリード線10数本が取れてしまって，もとに戻せなくなってしまった。そこで彼はついにあきらめてしまい，毎日カラーCRTに表示された「Make ready CMT」という一文を眺めては，こう叫んでいる「シャレにならんでー！」　木藤　弘志（16）山口県

```
0271: A1B2 19                  ADD    HL,DE
0272: A1B3 110099              LD     DE,ODATA
0273: A1B6 19                  ADD    HL,DE
0274: A1B7 2261AC              LD     (DATADR),HL
0275: A1BA 3E04                LD     A,4
0276: A1BC 3265AC              LD     (OCT),A
0277: A1BF DD360007            LD     (IX+0),7
0278: A1C3 DD360114            LD     (IX+1),20
0279: A1C7 3E34                LD     A,34H
0280: A1C9 CD0BA0              CALL   #PRINT
0281: A1CC DD360009            LD     (IX+0),9
0282: A1D0 DD360114            LD     (IX+1),20
0283: A1D4 3E2E                LD     A,CHRTEN
0284: A1D6 CD0BA0              CALL   #PRINT
0285: A1D9 3E30                LD     A,CHRZER
0286: A1DB CD0BA0              CALL   #PRINT
0287: A1DE AF                  XOR    A
0288: A1DF 325EAC              LD     (CURXS),A
0289: A1E2 325FAC              LD     (CURYS),A
0290: A1E5 3263AC              LD     (NOBASU),A
0291: A1E8 3266AC              LD     (PARTNO),A
0292: A1EB F5         CRTML1:  PUSH   AF              ;KAKU DATA PRINT&WRITE
0293: A1EC CD21AA              CALL   DATOUT
0294: A1EF 00                  NOP
0295: A1F0 47                  LD     B,A
0296: A1F1 CD19A9              CALL   DATINP
0297: A1F4 F1                  POP    AF
0298: A1F5 3C                  INC    A
0299: A1F6 325FAC              LD     (CURYS),A
0300: A1F9 FE0F                CP     15
0301: A1FB 20EE                JR     NZ,CRTML1
0302: A1FD 0601                LD     B,1             ;KAKU DATA PRINT&WRITE 2
0303: A1FF 78         CRTML3:  LD     A,B
0304: A200 325EAC              LD     (CURXS),A
0305: A203 0E02                LD     C,2
0306: A205 79         CRTML2:  LD     A,C
0307: A206 325FAC              LD     (CURYS),A
0308: A209 C5                  PUSH   BC
0309: A20A CD21AA              CALL   DATOUT
0310: A20D 00                  NOP
0311: A20E 47                  LD     B,A
0312: A20F CD19A9              CALL   DATINP
0313: A212 C1                  POP    BC
0314: A213 0C                  INC    C
0315: A214 3E0B                LD     A,11
0316: A216 B9                  CP     C
0317: A217 20EC                JR     NZ,CRTML2
0318: A219 04                  INC    B
0319: A21A 3E04                LD     A,4
0320: A21C B8                  CP     B
0321: A21D 20E0                JR     NZ,CRTML3
0322: A21F 3A67AC              LD     A,(DATNO)       ;DATADR KEISAN&N.PRINT
0323: A222 DD360004            LD     (IX+0),4
0324: A226 DD360101            LD     (IX+1),1
0325: A22A CDDDA8              CALL   PR10
0326: A22D 3E0F                LD     A,15
0327: A22F DD7700              LD     (IX+0),A
0328: A232 3E01                LD     A,1
0329: A234 DD7701              LD     (IX+1),A
0330: A237 ED5B61AC            LD     DE,(DATADR)
0331: A23B CD11A0              CALL   #MSG
0332: A23E AF                  XOR    A
0333: A23F 325EAC              LD     (CURXS),A
0334: A242 325FAC              LD     (CURYS),A
0335: A245 CD94A8              CALL   CURSET
0336: A248            ;
0337: A248            ; KEY INPUT
0338: A248            ;
0339: A248 CD0EA0     INKEY:   CALL   #GETKY
0340: A24B FE00                CP     0
0341: A24D 28F9                JR     Z,INKEY
0342: A24F 3264AC              LD     (KEYB),A
0343: A252 21BCAC              LD     HL,KEYDAT
0344: A255 012000              LD     BC,KEYSUU
0345: A258 EDB9                CPDR
0346: A25A 20EC                JR     NZ,INKEY
0347: A25C 79                  LD     A,C
0348: A25D FE11                CP     NOTEKY
0349: A25F DA05A3              JP     C,KYNOTE
0350: A262 D611                SUB    NOTEKY
0351: A264 1600                LD     D,0
0352: A266 5F                  LD     E,A
0353: A267 87                  ADD    A,A
0354: A268 83                  ADD    A,E
0355: A269 5F                  LD     E,A
0356: A26A 216FA2              LD     HL,KYJPT
0357: A26D 19                  ADD    HL,DE
0358: A26E E9                  JP     (HL)
0359: A26F C30AA4     KYJPT:   JP     KYOUP
0360: A272 C306A4              JP     KYODW
0361: A275 C32BA4              JP     KYDUP
0362: A278 C327A4              JP     KYDDW
0363: A27B C333A4              JP     KYDHUP
0364: A27E C32FA4              JP     KYDHDW
0365: A281 C3D2A7              JP     KYCLF
0366: A284 C347A8              JP     KYCDW
0367: A287 C3F8A7              JP     KYCRI
0368: A28A C31DA8              JP     KYCUP
0369: A28D C37AA4              JP     KYCR
0370: A290 C3D3A2              JP     KYSPC
0371: A293 C39CA2              JP     KYNOBA
0372: A296 C35BA4              JP     KYNEUP
0373: A299 C36AA4              JP     KYNEDW
0374: A29C            ;
0375: A29C            ; KEY NOBASU/NOBASANAI CH
0376: A29C            ;
0377: A29C DD360009   KYNOBA:  LD     (IX+0),9
0378: A2A0 DD360114            LD     (IX+1),20
0379: A2A4 3A63AC              LD     A,(NOBASU)
0380: A2A7 3C                  INC    A
0381: A2A8 FE04                CP     4
0382: A2AA 2001                JR     NZ,KYNBL3
0383: A2AC AF                  XOR    A
0384: A2AD 3263AC     KYNBL3:  LD     (NOBASU),A
0385: A2B0 E601                AND    1
0386: A2B2 2804                JR     Z,KYNBL1
0387: A2B4 3E2D                LD     A,CHRMIN
0388: A2B6 1802                JR     KYNBL2
0389: A2B8 3E2E       KYNBL1:  LD     A,CHRTEN
0390: A2BA CD0BA0     KYNBL2:  CALL   #PRINT
0391: A2BD 3A63AC              LD     A,(NOBASU)
0392: A2C0 E602                AND    2
0393: A2C2 2804                JR     Z,KYNBL4
0394: A2C4 3E2B                LD     A,CHRPUR
0395: A2C6 1802                JR     KYNBL5
0396: A2C8 3E30       KYNBL4:  LD     A,CHRZER
0397: A2CA CD0BA0     KYNBL5:  CALL   #PRINT
0398: A2CD CD71A8              CALL   GETKWT
0399: A2D0 C348A2              JP     INKEY
0400: A2D3            ;
0401: A2D3            ; NOBASU STOP
0402: A2D3            ;
0403: A2D3 1600       KYSPC:   LD     D,0
0404: A2D5 F3                  DI
0405: A2D6 010000     KYSPC1:  LD     BC,OPN#0
0406: A2D9 3E28                LD     A,KESADR
0407: A2DB ED79                OUT    (C),A
0408: A2DD 00                  NOP
0409: A2DE 03                  INC    BC
0410: A2DF 7A                  LD     A,D
0411: A2E0 00                  NOP
0412: A2E1 00                  NOP
0413: A2E2 ED79                OUT    (C),A
0414: A2E4 00                  NOP
0415: A2E5 010200              LD     BC,OPN#1
0416: A2E8 3E28                LD     A,KESADR
0417: A2EA ED79                OUT    (C),A
0418: A2EC 00                  NOP
0419: A2ED 03                  INC    BC
0420: A2EE 7A                  LD     A,D
0421: A2EF 00                  NOP
0422: A2F0 00                  NOP
0423: A2F1 ED79                OUT    (C),A
0424: A2F3 00                  NOP
0425: A2F4 14                  INC    D
0426: A2F5 3E03                LD     A,3
0427: A2F7 00                  NOP
0428: A2F8 00                  NOP
0429: A2F9 00                  NOP
0430: A2FA BA                  CP     D
0431: A2FB 20D9                JR     NZ,KYSPC1
0432: A2FD FB                  EI
0433: A2FE AF                  XOR    A
0434: A2FF 3266AC              LD     (PARTNO),A
0435: A302 C348A2              JP     INKEY
0436: A305            ;
0437: A305            ; NOTE KEY ON
0438: A305            ;
0439: A305 87         KYNOTE:  ADD    A,A
0440: A306 1600                LD     D,0
0441: A308 5F                  LD     E,A
0442: A309 21BDAC              LD     HL,NOTDAT
0443: A30C 19                  ADD    HL,DE
0444: A30D 5E                  LD     E,(HL)
0445: A30E 23                  INC    HL
0446: A30F 56                  LD     D,(HL)
0447: A310 3A65AC              LD     A,(OCT)         ;OCT DATA ADD
0448: A313 87                  ADD    A,A
0449: A314 87                  ADD    A,A
0450: A315 87                  ADD    A,A
0451: A316 B2                  OR     D
0452: A317 57                  LD     D,A
0453: A318 2600                LD     H,0
0454: A31A 3A66AC              LD     A,(PARTNO)
0455: A31D 87                  ADD    A,A
0456: A31E 6F                  LD     L,A
0457: A31F 0107AC              LD     BC,NOTE
0458: A322 09                  ADD    HL,BC
0459: A323 73                  LD     (HL),E
0460: A324 23                  INC    HL
0461: A325 72                  LD     (HL),D
0462: A326 3A66AC              LD     A,(PARTNO)
0463: A329 CB47                BIT    0,A
0464: A32B 2805                JR     Z,KYNOL2
0465: A32D 010200              LD     BC,OPN#1
0466: A330 1803                JR     KYNOL3
0467: A332 010000     KYNOL2:  LD     BC,OPN#0
0468: A335 CB3F       KYNOL3:  SRL    A
0469: A337 C6A4                ADD    A,FNHSAD
0470: A339 F3                  DI
0471: A33A ED79                OUT    (C),A
0472: A33C 00                  NOP
0473: A33D 03                  INC    BC
0474: A33E 7A                  LD     A,D
0475: A33F 00                  NOP
0476: A340 00                  NOP
0477: A341 ED79                OUT    (C),A
0478: A343 00                  NOP
0479: A344 0B                  DEC    BC
0480: A345 3A66AC              LD     A,(PARTNO)
0481: A348 CB3F                SRL    A
0482: A34A C6A0                ADD    A,FNLSAD
0483: A34C 0000                DEFW   0000H           ;NOP*18
0484: A34E 0000                DEFW   0000H
0485: A350 0000                DEFW   0000H
0486: A352 0000                DEFW   0000H
0487: A354 0000                DEFW   0000H
0488: A356 0000                DEFW   0000H
0489: A358 0000                DEFW   0000H
0490: A35A 0000                DEFW   0000H
0491: A35C 0000                DEFW   0000H
0492: A35E ED79                OUT    (C),A
0493: A360 00                  NOP
0494: A361 03                  INC    BC
0495: A362 7B                  LD     A,E
0496: A363 00                  NOP
0497: A364 00                  NOP
0498: A365 ED79                OUT    (C),A
0499: A367 00                  NOP
0500: A368 3A66AC              LD     A,(PARTNO)
0501: A36B 1600                LD     D,0
0502: A36D 5F                  LD     E,A
0503: A36E AF                  XOR    A
0504: A36F 2113AC              LD     HL,DEEPS
0505: A372 19                  ADD    HL,DE
0506: A373 77                  LD     (HL),A
0507: A374 2125AC              LD     HL,LFOPS
0508: A377 19                  ADD    HL,DE
0509: A378 77                  LD     (HL),A
0510: A379 3C                  INC    A
0511: A37A 211FAC              LD     HL,LFOUD
0512: A37D 19                  ADD    HL,DE
0513: A37E 77                  LD     (HL),A
0514: A37F 2A61AC              LD     HL,(DATADR)     ;S.WAIT PUT
0515: A382 112500              LD     DE,SWTOF
0516: A385 19                  ADD    HL,DE
0517: A386 C5                  PUSH   BC
0518: A387 46                  LD     B,(HL)
0519: A388 3A66AC              LD     A,(PARTNO)
0520: A38B 1600                LD     D,0
0521: A38D 5F                  LD     E,A
0522: A38E 2119AC              LD     HL,STIMES
0523: A391 19                  ADD    HL,DE
0524: A392 70                  LD     (HL),B
0525: A393 C1                  POP    BC
0526: A394 0B                  DEC    BC
0527: A395 3E28                LD     A,KESADR
0528: A397 ED79                OUT    (C),A
0529: A399 00                  NOP
0530: A39A 03                  INC    BC
0531: A39B 3A63AC              LD     A,(NOBASU)
0532: A39E E601                AND    1
0533: A3A0 282B                JR     Z,KYNOL1
0534: A3A2 3A66AC              LD     A,(PARTNO)      ;NOBASU------------
0535: A3A5 CB3F                SRL    A
0536: A3A7 ED79                OUT    (C),A           ;SLOT ALL OFF
0537: A3A9 00                  NOP
0538: A3AA 3A66AC              LD     A,(PARTNO)
0539: A3AD CB3F                SRL    A
0540: A3AF C6F0                ADD    A,SALLON        ;SLOT ALL ON
0541: A3B1 0000                DEFW   0000H           ;NOP*19
0542: A3B3 0000                DEFW   0000H
0543: A3B5 0000                DEFW   0000H
0544: A3B7 0000                DEFW   0000H
0545: A3B9 0000                DEFW   0000H
0546: A3BB 0000                DEFW   0000H
0547: A3BD 0000                DEFW   0000H
0548: A3BF 0000                DEFW   0000H
0549: A3C1 0000                DEFW   0000H
0550: A3C3 00                  DEFB   00H
```


▶先日，通学バスの中で聞いた女子高校生同士の会話。「あたしさあ，パソコンってゲームしかできないと思っていたけど，計算とかいろいろできるのネェ」。うっ嬉しい。こういう人がひとりでもふえてくれることは非常に嬉しい。しかし，よく考えれば高等学校数学IIには「電子計算機と流れ図」という単元があるんですよね。　埜口　秀人（16）茨城県

```
0551: A3C4 ED79              OUT   (C),A
0552: A3C6 00                NOP
0553: A3C7 FB                EI
0554: A3C8 CD7FA8            CALL  GETKWN
0555: A3CB 1822              JR    KYNOL4
0556: A3CD 3A66AC   KYNOL1:  LD    A,(PARTNO)        ;SLOT ON
0557: A3D0 CB3F              SRL   A
0558: A3D2 C6F0              ADD   A,SALLON          ;SLOT ALL ON
0559: A3D4 ED79              OUT   (C),A
0560: A3D6 00                NOP
0561: A3D7 FB                EI
0562: A3D8 CD7FA8            CALL  GETKWN
0563: A3DB F3                DI
0564: A3DC 0B                DEC   BC
0565: A3DD 3E28              LD    A,KESADR
0566: A3DF ED79              OUT   (C),A
0567: A3E1 00                NOP
0568: A3E2 00                NOP
0569: A3E3 00                NOP
0570: A3E4 00                NOP
0571: A3E5 03                INC   BC
0572: A3E6 3A66AC            LD    A,(PARTNO)
0573: A3E9 CB3F              SRL   A
0574: A3EB ED79              OUT   (C),A             ;SLOT ALL OFF
0575: A3ED 00                NOP
0576: A3EE FB                EI
0577: A3EF 3A63AC   KYNOL4:  LD    A,(NOBASU)
0578: A3F2 E602              AND   2
0579: A3F4 CA48A2            JP    Z,INKEY
0580: A3F7 3A66AC            LD    A,(PARTNO)
0581: A3FA 3C                INC   A
0582: A3FB FE06              CP    6
0583: A3FD 2001              JR    NZ,KYNOL5
0584: A3FF AF                XOR   A
0585: A400 3266AC   KYNOL5:  LD    (PARTNO),A
0586: A403 C348A2            JP    INKEY
0587: A406                   ;
0588: A406                   ; OCT UP DOWN KEY
0589: A406                   ;
0590: A406 3EFF     KYODW:   LD    A,FFH             ;-1:DOWN
0591: A408 1802              JR    KYOL1
0592: A40A 3E01     KYOUP:   LD    A,1               ;+1:UP
0593: A40C 2165AC   KYOL1:   LD    HL,OCT
0594: A40F 86                ADD   A,(HL)
0595: A410 E607              AND   07H               ;00000111B
0596: A412 77                LD    (HL),A
0597: A413 DD360007          LD    (IX+0),7
0598: A417 DD360114          LD    (IX+1),20
0599: A41B 3E30              LD    A,CHRZER
0600: A41D 86                ADD   A,(HL)
0601: A41E CD0BA0            CALL  #PRINT
0602: A421 CD71A8            CALL  GETKWT
0603: A424 C348A2            JP    INKEY
0604: A427                   ;
0605: A427                   ; DATA UP DOWN KEY
0606: A427                   ;
0607: A427 06FF     KYDDW:   LD    B,FFH             ;-1:DOWN
0608: A429 180A              JR    KYD1
0609: A42B 0601     KYDUP:   LD    B,1               ;+1:UP
0610: A42D 1806              JR    KYD1
0611: A42F 06FB     KYDHDW:  LD    B,FBH             ;-5:HIGH DOWN
0612: A431 1802              JR    KYD1
0613: A433 0605     KYDHUP:  LD    B,5               ;+5:HIGH UP
0614: A435 2170AC   KYD1:    LD    HL,DATDAT
0615: A438 3A5FAC            LD    A,(CURYS)
0616: A43B 1600              LD    D,0
0617: A43D 5F                LD    E,A
0618: A43E 87                ADD   A,A
0619: A43F 83                ADD   A,E
0620: A440 5F                LD    E,A
0621: A441 19                ADD   HL,DE
0622: A442 7E                LD    A,(HL)
0623: A443 CB3F     KYDL1:   SRL   A
0624: A445 3804              JR    C,KYD2
0625: A447 CB20              SLA   B
0626: A449 18F8              JR    KYDL1
0627: A44B C5       KYD2:    PUSH  BC
0628: A44C CD21AA            CALL  DATOUT
0629: A44F C1                POP   BC
0630: A450 80                ADD   A,B
0631: A451 47                LD    B,A
0632: A452 CD19A9            CALL  DATINP
0633: A455 CD8BA8            CALL  KWAIT
0634: A458 C348A2            JP    INKEY
0635: A45B                   ;
0636: A45B                   ; NEIRO DATA UP
0637: A45B 3A67AC   KYNEUP:  LD    A,(DATNO)
0638: A45E 3C                INC   A
0639: A45F FE26              CP    DATASU
0640: A461 2001              JR    NZ,KYNUL1
0641: A463 AF                XOR   A
0642: A464 3267AC   KYNUL1:  LD    (DATNO),A
0643: A467 C3ECA0            JP    CRTMAK
0644: A46A                   ;
0645: A46A                   ; NEIRO DATA DOWN
0646: A46A                   ;
0647: A46A 3A67AC   KYNEDW:  LD    A,(DATNO)
0648: A46D 3D                DEC   A
0649: A46E FEFF              CP    -1
0650: A470 2002              JR    NZ,KYNDL1
0651: A472 3E25              LD    A,37
0652: A474 3267AC   KYNDL1:  LD    (DATNO),A
0653: A477 C3ECA0            JP    CRTMAK
0654: A47A                   ;
0655: A47A                   ; CR KEY
0656: A47A                   ;
0657: A47A CDAAA8   KYCR:    CALL  CURCLR
0658: A47D DD360016          LD    (IX+0),22
0659: A481 DD360110          LD    (IX+1),16
0660: A485 11B1AD            LD    DE,M22
0661: A488 CD11A0            CALL  #MSG
0662: A48B DD360016          LD    (IX+0),22
0663: A48F DD360111          LD    (IX+1),17
0664: A493 11ACAD            LD    DE,M21
0665: A496 CD11A0            CALL  #MSG
0666: A499 DD360016          LD    (IX+0),22
0667: A49D DD360112          LD    (IX+1),18
0668: A4A1 11B7AD            LD    DE,M23
0669: A4A4 CD11A0            CALL  #MSG
0670: A4A7 DD360016          LD    (IX+0),22
0671: A4AB DD360113          LD    (IX+1),19
0672: A4AF 11BCAD            LD    DE,M24
0673: A4B2 CD11A0            CALL  #MSG
0674: A4B5 DD360016          LD    (IX+0),22
0675: A4B9 DD360114          LD    (IX+1),20
0676: A4BD 11C1AD            LD    DE,M25
0677: A4C0 CD11A0            CALL  #MSG
0678: A4C3 DD360016          LD    (IX+0),22
0679: A4C7 DD360115          LD    (IX+1),21
0680: A4CB 11C6AD            LD    DE,M30
0681: A4CE CD11A0            CALL  #MSG
0682: A4D1 0610              LD    B,16
0683: A4D3 CD71A8            CALL  GETKWT
0684: A4D6 CD8BA8   KYCRL5:  CALL  KWAIT
0685: A4D9 DD360014          LD    (IX+0),20
0686: A4DD DD7001            LD    (IX+1),B
0687: A4E0 3E3E              LD    A,CHRDAI
0688: A4E2 CD0BA0            CALL  #PRINT
0689: A4E5 CD0EA0            CALL  #GETKY
0690: A4E8 FE4B              CP    CHRK
0691: A4EA 280A              JR    Z,KYCRL1
0692: A4EC FE2C              CP    CHRCON
0693: A4EE 281C              JR    Z,KYCRL2
0694: A4F0 FE0D              CP    CHRCR
0695: A4F2 282E              JR    Z,KYCRL3
0696: A4F4 18E0              JR    KYCRL5
0697: A4F6 DD360014 KYCRL1:  LD    (IX+0),20
0698: A4FA DD7001            LD    (IX+1),B
0699: A4FD 3E20              LD    A,CHRSPC
0700: A4FF CD0BA0            CALL  #PRINT
0701: A502 05                DEC   B
0702: A503 3E0F              LD    A,15
0703: A505 B8                CP    B
0704: A506 20CE              JR    NZ,KYCRL5
0705: A508 0614              LD    B,20
0706: A50A 18CA              JR    KYCRL5
0707: A50C DD360014 KYCRL2:  LD    (IX+0),20
0708: A510 DD7001            LD    (IX+1),B
0709: A513 3E20              LD    A,CHRSPC
0710: A515 CD0BA0            CALL  #PRINT
0711: A518 04                INC   B
0712: A519 3E16              LD    A,22
0713: A51B B8                CP    B
0714: A51C 20B8              JR    NZ,KYCRL5
0715: A51E 0610              LD    B,16
0716: A520 18B4              JR    KYCRL5
0717: A522                   ;
0718: A522 78       KYCRL3:  LD    A,B
0719: A523 FE11              CP    17
0720: A525 CA4BA5            JP    Z,CRNAME
0721: A528 FE10              CP    16
0722: A52A CA98A5            JP    Z,CRNEIR
0723: A52D FE12              CP    18
0724: A52F CA79A7            JP    Z,CRLOAD
0725: A532 FE13              CP    19
0726: A534 CA17A7            JP    Z,CRSAVE
0727: A537 FE14              CP    20
0728: A539 CAAEA6            JP    Z,CRCOPY
0729: A53C C308A0            JP    EXIT
0730: A53F                   ;
0731: A53F CD97A7   KYCRLE:  CALL  CRCLR
0732: A542 CD71A8            CALL  GETKWT
0733: A545 CD94A8            CALL  CURSET
0734: A548 C348A2            JP    INKEY
0735: A54B                   ;
0736: A54B DD360015 CRNAME:  LD    (IX+0),21         ;NAME INPUT
0737: A54F DD360115          LD    (IX+1),21
0738: A553 11ACAD            LD    DE,M21
0739: A556 CD11A0            CALL  #MSG
0740: A559 CD71A8            CALL  GETKWT
0741: A55C DD360116          LD    (IX+1),22
0742: A560 CDADA7            CALL  GET1L
0743: A563 DA3FA5            JP    C,KYCRLE
0744: A566 D5                PUSH  DE
0745: A567 DD36000F          LD    (IX+0),15
0746: A56B DD360101          LD    (IX+1),1
0747: A56F 11D3AD            LD    DE,M27
0748: A572 CD11A0            CALL  #MSG
0749: A575 D1                POP   DE
0750: A576 DD36000F          LD    (IX+0),15
0751: A57A DD360101          LD    (IX+1),1
0752: A57E 2A61AC            LD    HL,(DATADR)
0753: A581 060A              LD    B,10
0754: A583 AF       CRNAL1:  XOR   A
0755: A584 B9                CP    C
0756: A585 3E20              LD    A,CHRSPC
0757: A587 2801              JR    Z,CRNAL2
0758: A589 1A                LD    A,(DE)
0759: A58A 77       CRNAL2:  LD    (HL),A
0760: A58B CD0BA0            CALL  #PRINT
0761: A58E 0D                DEC   C
0762: A58F 23                INC   HL
0763: A590 13                INC   DE
0764: A591 10F0              DJNZ  CRNAL1
0765: A593 360D              LD    (HL),CHRCR
0766: A595 C33FA5            JP    KYCRLE
0767: A598                   ;
0768: A598 CDA5A5   CRNEIR:  CALL  NEICRT
0769: A59B CD49A6            CALL  CRNECU
0770: A59E 78                LD    A,B
0771: A59F 3267AC            LD    (DATNO),A
0772: A5A2 C3ECA0            JP    CRTMAK
0773: A5A5                   ;
0774: A5A5 3E0C     NEICRT:  LD    A,CHRCLS          ;NEIRO CRT
0775: A5A7 CD0BA0            CALL  #PRINT
0776: A5AA DD360010          LD    (IX+0),16
0777: A5AE DD360101          LD    (IX+1),1
0778: A5B2 11ECAD            LD    DE,M29
0779: A5B5 CD11A0            CALL  #MSG
0780: A5B8 DD360011          LD    (IX+0),17
0781: A5BC DD360102          LD    (IX+1),2
0782: A5C0 11B1AD            LD    DE,M22
0783: A5C3 CD11A0            CALL  #MSG
0784: A5C6 DD360010          LD    (IX+0),16
0785: A5CA DD360103          LD    (IX+1),3
0786: A5CE 11ECAD            LD    DE,M29
0787: A5D1 CD11A0            CALL  #MSG
0788: A5D4 DD360005          LD    (IX+0),5
0789: A5D8 DD360104          LD    (IX+1),4
0790: A5DC 11DFAC            LD    DE,M1
0791: A5DF CD11A0            CALL  #MSG
0792: A5E2 DD36000A          LD    (IX+0),10
0793: A5E6 11ACAD            LD    DE,M21
0794: A5E9 CD11A0            CALL  #MSG
0795: A5EC DD360016          LD    (IX+0),22
0796: A5F0 11DFAC            LD    DE,M1
0797: A5F3 CD11A0            CALL  #MSG
0798: A5F6 DD36001B          LD    (IX+0),27
0799: A5FA 11ACAD            LD    DE,M21
0800: A5FD CD11A0            CALL  #MSG
0801: A600 0600              LD    B,0
0802: A602 DD360105          LD    (IX+1),5
0803: A606 110099            LD    DE,ODATA
0804: A609 21F802            LD    HL,ODMID
0805: A60C 19                ADD   HL,DE
0806: A60D EB                EX    DE,HL
0807: A60E DD360004 CRNEL1:  LD    (IX+0),4
0808: A612 78                LD    A,B
0809: A613 CDDDA8            CALL  PR10
0810: A616 3E3A              LD    A,CHRCOR
0811: A618 CD0BA0            CALL  #PRINT
0812: A61B EB                EX    DE,HL
0813: A61C CD11A0            CALL  #MSG
0814: A61F DD360015          LD    (IX+0),21
0815: A623 3E13              LD    A,19
0816: A625 80                ADD   A,B
0817: A626 CDDDA8            CALL  PR10
0818: A629 3E3A              LD    A,CHRCOR
0819: A62B CD0BA0            CALL  #PRINT
0820: A62E EB                EX    DE,HL
0821: A62F CD11A0            CALL  #MSG
0822: A632 D5                PUSH  DE
0823: A633 112800            LD    DE,DATALO
0824: A636 19                ADD   HL,DE
0825: A637 D1                POP   DE
0826: A638 E5                PUSH  HL
0827: A639 212800            LD    HL,DATALO
0828: A63C 19                ADD   HL,DE
0829: A63D EB                EX    DE,HL
0830: A63E E1                POP   HL
```

▶中学時代以来，何年もハンダゴテを握ったことのない私にとって，今回のハードウェア入門はとてもなつかしいもののように思えました。さっそくこの“ハードS-OS”の企画にも参加しようと思っています。　　工藤　昭彦（22）東京都

```
0831: A63F 04                 INC   B
0832: A640 CD14A0             CALL  #LTNL
0833: A643 3E13               LD    A,19
0834: A645 B8                 CP    B
0835: A646 20C6               JR    NZ,CRNEL1
0836: A648 C9                 RET
0837: A649           ;
0838: A649 3A67AC    CRNECU:  LD    A,(DATNO)
0839: A64C 47                 LD    B,A
0840: A64D CD71A8             CALL  GETKWT
0841: A650 CD6BA6    CRNEL4:  CALL  CRNEL9
0842: A653 3E3E               LD    A,CHRDAI
0843: A655 CD0BA0             CALL  #PRINT
0844: A658 CD8BA8             CALL  KWAIT
0845: A65B CD0EA0    CRNEL8:  CALL  #GETKY
0846: A65E FE4B               CP    CHRK
0847: A660 2824               JR    Z,CRNEL5
0848: A662 FE2C               CP    CHRCON
0849: A664 2834               JR    Z,CRNEL6
0850: A666 FE0D               CP    CHRCR
0851: A668 C8                 RET   Z
0852: A669 18F0               JR    CRNEL8
0853: A66B 3E12      CRNEL9:  LD    A,18              ;CURSOR SET
0854: A66D B8                 CP    B
0855: A66E 3808               JR    C,CRNELA
0856: A670 3E05               LD    A,5
0857: A672 80                 ADD   A,B
0858: A673 DD360004           LD    (IX+0),4
0859: A677 DD7701             LD    (IX+1),A
0860: A67A C9                 RET
0861: A67B 3EF2      CRNELA:  LD    A,-14
0862: A67D 80                 ADD   A,B
0863: A67E DD360015           LD    (IX+0),21
0864: A682 DD7701             LD    (IX+1),A
0865: A685 C9                 RET
0866: A686           ;
0867: A686 CD6BA6    CRNEL5:  CALL  CRNEL9            ;UP
0868: A689 3E20               LD    A,CHRSPC
0869: A68B CD0BA0             CALL  #PRINT
0870: A68E 05                 DEC   B
0871: A68F 3EFF               LD    A,-1
0872: A691 B8                 CP    B
0873: A692 C250A6             JP    NZ,CRNEL4
0874: A695 0625               LD    B,37
0875: A697 C350A6             JP    CRNEL4
0876: A69A           ;
0877: A69A CD6BA6    CRNEL6:  CALL  CRNEL9            ;DOWN
0878: A69D 3E20               LD    A,CHRSPC
0879: A69F CD0BA0             CALL  #PRINT
0880: A6A2 04                 INC   B
0881: A6A3 3E26               LD    A,38
0882: A6A5 B8                 CP    B
0883: A6A6 C250A6             JP    NZ,CRNEL4
0884: A6A9 0600               LD    B,0
0885: A6AB C350A6             JP    CRNEL4
0886: A6AE           ;
0887: A6AE CDA5A5    CRCOPY:  CALL  NEICRT            ;NEIRO COPY
0888: A6B1 DD360005           LD    (IX+0),5
0889: A6B5 DD360100           LD    (IX+1),0
0890: A6B9 11C6AD             LD    DE,M30
0891: A6BC CD11A0             CALL  #MSG
0892: A6BF DD360007           LD    (IX+0),7
0893: A6C3 DD360101           LD    (IX+1),1
0894: A6C7 11CBAD             LD    DE,M31
0895: A6CA CD11A0             CALL  #MSG
0896: A6CD CD49A6             CALL  CRNECU
0897: A6D0 78                 LD    A,B
0898: A6D1 3267AC             LD    (DATNO),A
0899: A6D4 DD360004           LD    (IX+0),4
0900: A6D8 DD360101           LD    (IX+1),1
0901: A6DC CDDDA8             CALL  PR10
0902: A6DF CD49A6             CALL  CRNECU
0903: A6E2 3A67AC             LD    A,(DATNO)
0904: A6E5 4F                 LD    C,A
0905: A6E6 78                 LD    A,B
0906: A6E7 3267AC             LD    (DATNO),A
0907: A6EA 79                 LD    A,C
0908: A6EB 6F                 LD    L,A
0909: A6EC 2600               LD    H,0
0910: A6EE 29                 ADD   HL,HL             ;TENSOU DATA ADR.
0911: A6EF 29                 ADD   HL,HL
0912: A6F0 29                 ADD   HL,HL
0913: A6F1 E5                 PUSH  HL
0914: A6F2 29                 ADD   HL,HL
0915: A6F3 29                 ADD   HL,HL
0916: A6F4 D1                 POP   DE
0917: A6F5 19                 ADD   HL,DE
0918: A6F6 110099             LD    DE,ODATA
0919: A6F9 19                 ADD   HL,DE
0920: A6FA E5                 PUSH  HL
0921: A6FB 3A67AC             LD    A,(DATNO)         ;TENSOU SAKI DATA ADR.
0922: A6FE 6F                 LD    L,A
0923: A6FF 2600               LD    H,0
0924: A701 29                 ADD   HL,HL
0925: A702 29                 ADD   HL,HL
0926: A703 29                 ADD   HL,HL
0927: A704 E5                 PUSH  HL
0928: A705 29                 ADD   HL,HL
0929: A706 29                 ADD   HL,HL
0930: A707 D1                 POP   DE
0931: A708 19                 ADD   HL,DE
0932: A709 110099             LD    DE,ODATA
0933: A70C 19                 ADD   HL,DE
0934: A70D D1                 POP   DE
0935: A70E EB                 EX    DE,HL
0936: A70F 012800             LD    BC,40
0937: A712 EDB0               LDIR
0938: A714 C3ECA0             JP    CRTMAK
0939: A717           ;
0940: A717 CD97A7    CRSAVE:  CALL  CRCLR             ;DATA SAVE
0941: A71A DD360015           LD    (IX+0),21
0942: A71E DD360110           LD    (IX+1),16
0943: A722 11E7AD             LD    DE,M28
0944: A725 CD11A0             CALL  #MSG
0945: A728 CD1AA0             CALL  #PRNTS
0946: A72B 11ACAD             LD    DE,M21
0947: A72E CD11A0             CALL  #MSG
0948: A731 CD71A8             CALL  GETKWT
0949: A734 DD360111           LD    (IX+1),17
0950: A738 CDADA7             CALL  GET1L
0951: A73B DA3FA5             JP    C,KYCRLE
0952: A73E 3E01               LD    A,OBJINF
0953: A740 CD2CA0             CALL  #FILE
0954: A743 FD2A741F           LD    IY,(#IBFAD)
0955: A747 FD36110D           LD    (IY+17),0DH
0956: A74B 210099             LD    HL,ODATA
0957: A74E 22701F             LD    (#DTADR),HL
0958: A751 21F005             LD    HL,ODATAS
0959: A754 22721F             LD    (#SIZE),HL
0960: A757 DD360015           LD    (IX+0),21
0961: A75B DD360114           LD    (IX+1),20
0962: A75F CD20A0             CALL  #WRI
0963: A762 DA3FA5             JP    C,KYCRLE
0964: A765 CD23A0             CALL  #WRD
0965: A768 DD360000           LD    (IX+0),0
0966: A76C DD360115           LD    (IX+1),21
0967: A770 11D3AD             LD    DE,M27
0968: A773 CD11A0             CALL  #MSG
0969: A776 C33FA5             JP    KYCRLE
0970: A779           ;
```

```
0971: A779 CD97A7    CRLOAD:  CALL  CRCLR
0972: A77C DD360015           LD    (IX+0),21
0973: A780 DD360110           LD    (IX+1),16
0974: A784 CD26A0             CALL  #RDI
0975: A787 DA3FA5             JP    C,KYCRLE
0976: A78A DD360015           LD    (IX+0),21
0977: A78E CD2FA0             CALL  #FPRNT
0978: A791 CD29A0             CALL  #RDD
0979: A794 C39BA5             JP    CRNEIR
0980: A797           ;
0981: A797 0608      CRCLR:   LD    B,8
0982: A799 DD360110           LD    (IX+1),16
0983: A79D DD360014  CRCLR1:  LD    (IX+0),20
0984: A7A1 11D3AD             LD    DE,M27
0985: A7A4 CD11A0             CALL  #MSG
0986: A7A7 CD14A0             CALL  #LTNL
0987: A7AA 10F1               DJNZ  CRCLR1
0988: A7AC C9                 RET
0989: A7AD           ;
0990: A7AD DD360015  GET1L:   LD    (IX+0),21
0991: A7B1 21761F             LD    HL,#KBFAD
0992: A7B4 5E                 LD    E,(HL)
0993: A7B5 23                 INC   HL
0994: A7B6 56                 LD    D,(HL)
0995: A7B7 CD1DA0             CALL  #GETL
0996: A7BA 1A                 LD    A,(DE)
0997: A7BB FE1B               CP    CHRBRK
0998: A7BD 2002               JR    NZ,GET1L1
0999: A7BF 37                 SCF
1000: A7C0 C9                 RET
1001: A7C1 211500    GET1L1:  LD    HL,21
1002: A7C4 19                 ADD   HL,DE
1003: A7C5 E5                 PUSH  HL
1004: A7C6 AF                 XOR   A
1005: A7C7 4F                 LD    C,A
1006: A7C8 BE        GET1L3:  CP    (HL)
1007: A7C9 2804               JR    Z,GET1L2
1008: A7CB 23                 INC   HL
1009: A7CC 0C                 INC   C
1010: A7CD 18F9               JR    GET1L3
1011: A7CF D1        GET1L2:  POP   DE
1012: A7D0 B7                 OR    A
1013: A7D1 C9                 RET
1014: A7D2           ;
1015: A7D2           ; CURSOR KEY
1016: A7D2           ;
1017: A7D2 3A5FAC    KYCLF:   LD    A,(CURYS)         ;LEFT
1018: A7D5 FE02               CP    2
1019: A7D7 DA4BA2             JP    C,INKEY
1020: A7DA FE0B               CP    11
1021: A7DC D24BA2             JP    NC,INKEY
1022: A7DF 3A5EAC             LD    A,(CURXS)
1023: A7E2 3D                 DEC   A
1024: A7E3 FEFF               CP    -1
1025: A7E5 2002               JR    NZ,KYCLL1
1026: A7E7 3E03               LD    A,3
1027: A7E9 CDAAA8    KYCLL1:  CALL  CURCLR
1028: A7EC 325EAC             LD    (CURXS),A
1029: A7EF CD94A8             CALL  CURSET
1030: A7F2 CD8BA8             CALL  KWAIT
1031: A7F5 C34BA2             JP    INKEY
1032: A7F8           ;
1033: A7F8 3A5FAC    KYCRI:   LD    A,(CURYS)         ;RIGHT
1034: A7FB FE02               CP    2
1035: A7FD DA4BA2             JP    C,INKEY
1036: A800 FE0B               CP    11
1037: A802 D24BA2             JP    NC,INKEY
1038: A805 3A5EAC             LD    A,(CURXS)
1039: A808 3C                 INC   A
1040: A809 FE04               CP    4
1041: A80B 2001               JR    NZ,KYCRI1
1042: A80D AF                 XOR   A
1043: A80E CDAAA8    KYCRI1:  CALL  CURCLR
1044: A811 325EAC             LD    (CURXS),A
1045: A814 CD94A8             CALL  CURSET
1046: A817 CD8BA8             CALL  KWAIT
1047: A81A C34BA2             JP    INKEY
1048: A81D           ;
1049: A81D CDAAA8    KYCUP:   CALL  CURCLR            ;UP
1050: A820 3A5FAC             LD    A,(CURYS)
1051: A823 3D                 DEC   A
1052: A824 FEFF               CP    -1
1053: A826 2004               JR    NZ,KYCUL1
1054: A828 3E0E               LD    A,14
1055: A82A 180F               JR    KYCUL2
1056: A82C FE01      KYCUL1:  CP    1
1057: A82E 200B               JR    NZ,KYCUL2
1058: A830 47                 LD    B,A
1059: A831 3A5EAC             LD    A,(CURXS)
1060: A834 FE01               CP    1
1061: A836 78                 LD    A,B
1062: A837 3802               JR    C,KYCUL2
1063: A839 3E0A               LD    A,10
1064: A83B 325FAC    KYCUL2:  LD    (CURYS),A
1065: A83E CD94A8             CALL  CURSET
1066: A841 CD8BA8             CALL  KWAIT
1067: A844 C34BA2             JP    INKEY
1068: A847           ;
1069: A847 CDAAA8    KYCDW:   CALL  CURCLR            ;DOWN
1070: A84A 3A5FAC             LD    A,(CURYS)
1071: A84D 3C                 INC   A
1072: A84E FE0F               CP    15
1073: A850 2003               JR    NZ,KYCDL1
1074: A852 AF                 XOR   A
1075: A853 1810               JR    KYCDL2
1076: A855 FE0B      KYCDL1:  CP    11
1077: A857 200C               JR    NZ,KYCDL2
1078: A859 47                 LD    B,A
1079: A85A 3A5EAC             LD    A,(CURXS)
1080: A85D FE01               CP    1
1081: A85F 78                 LD    A,B
1082: A860 DA65A8             JP    C,KYCDL2
1083: A863 3E02               LD    A,2
1084: A865 325FAC    KYCDL2:  LD    (CURYS),A
1085: A868 CD94A8             CALL  CURSET
1086: A86B CD8BA8             CALL  KWAIT
1087: A86E C34BA2             JP    INKEY
1088: A871           ;
1089: A871           ; OTOZUKURI NO SUB
1090: A871           ;
1091: A871           ;
1092: A871           ; KEY O HANASITARA RET
1093: A871           ;      BREAK  A,F
1094: A871           ;
1095: A871 C5        GETKWT:  PUSH  BC
1096: A872 0603      GETWL2:  LD    B,3
1097: A874 CD0EA0    GETWL1:  CALL  #GETKY
1098: A877 FE00               CP    0
1099: A879 20F7               JR    NZ,GETWL2
1100: A87B 10F7               DJNZ  GETWL1
1101: A87D C1                 POP   BC
1102: A87E C9                 RET
1103: A87F           ;
1104: A87F           ; KEY OFF THEN RETURN
1105: A87F           ;      BREAK  A,F
1106: A87F           ;
1107: A87F E5        GETKWN:  PUSH  HL
1108: A880 2164AC             LD    HL,KEYB
1109: A883 CD0EA0    GEWNL1:  CALL  #GETKY
1110: A886 BE                 CP    (HL)
```


▶この前，大阪の日本橋へ行って来た。スゴイですねぇ。人はいっぱい，電気屋さんもいっぱい，これから日本橋へ行ってみようと思っている皆さん，日本橋へ行くには近鉄の日本橋駅を使うより，地下鉄のえびす町駅で降りてください。駅から出るとそこは日本橋のド まん中です。
波多野　幸也（17）京都府

```
1111: A887 28FA                 JR     Z,GEWNL1
1112: A889 E1                   POP    HL
1113: A88A C9                   RET
1114: A88B              ;
1115: A88B              ; KEY  WAIT
1116: A88B              ;   BREAK HL,A
1117: A88B              ;
1118: A88B 217017       KWAIT:  LD     HL,KEYWSU
1119: A88E 2B           WAITL1: DEC    HL
1120: A88F 7C                   LD     A,H
1121: A890 B5                   OR     L
1122: A891 20FB                 JR     NZ,WAITL1
1123: A893 C9                   RET
1124: A894              ;
1125: A894              ; CURSOR SET
1126: A894              ;   BREAK  NO
1127: A894              ;
1128: A894 F5           CURSET: PUSH   AF
1129: A895 C5                   PUSH   BC
1130: A896 CDC0A8               CALL   CURKEI
1131: A899 3E5B                 LD     A,CHRDCO
1132: A89B CD0BA0               CALL   #PRINT
1133: A89E 78                   LD     A,B
1134: A89F DD7700               LD     (IX+0),A
1135: A8A2 3E5D                 LD     A,CHRDCC
1136: A8A4 CD0BA0               CALL   #PRINT
1137: A8A7 C1                   POP    BC
1138: A8A8 F1                   POP    AF
1139: A8A9 C9                   RET
1140: A8AA              ;
1141: A8AA              ; CURSOR CLEAR
1142: A8AA              ;   BREAK  NO
1143: A8AA              ;
1144: A8AA F5           CURCLR: PUSH   AF
1145: A8AB C5                   PUSH   BC
1146: A8AC CDC0A8               CALL   CURKEI
1147: A8AF 3E20                 LD     A,CHRSPC
1148: A8B1 CD0BA0               CALL   #PRINT
1149: A8B4 78                   LD     A,B
1150: A8B5 DD7700               LD     (IX+0),A
1151: A8B8 3E20                 LD     A,CHRSPC
1152: A8BA CD0BA0               CALL   #PRINT
1153: A8BD C1                   POP    BC
1154: A8BE F1                   POP    AF
1155: A8BF C9                   RET
1156: A8C0              ;
1157: A8C0              ; CURSOR KEISAN
1158: A8C0              ;   BREAK A,F,B
1159: A8C0              ;
1160: A8C0 3A5EAC       CURKEI: LD     A,(CURXS)
1161: A8C3 47                   LD     B,A
1162: A8C4 87                   ADD    A,A
1163: A8C5 87                   ADD    A,A
1164: A8C6 80                   ADD    A,B
1165: A8C7 C60A                 ADD    A,10
1166: A8C9 DD7700               LD     (IX+0),A
1167: A8CC C604                 ADD    A,4
1168: A8CE 47                   LD     B,A
1169: A8CF 3A5FAC               LD     A,(CURYS)
1170: A8D2 C603                 ADD    A,3
1171: A8D4 FE0E                 CP     14
1172: A8D6 3801                 JR     C,CURSL1
1173: A8D8 3C                   INC    A
1174: A8D9 DD7701       CURSL1: LD     (IX+1),A
1175: A8DC C9                   RET
1176: A8DD              ;
1177: A8DD              ; DECMAL 3KETA PRINT
1178: A8DD              ;   PRINT DATA ACC
1179: A8DD              ;   BREAK  A,F
1180: A8DD              ;
1181: A8DD C5           PR10:   PUSH   BC
1182: A8DE D5                   PUSH   DE
1183: A8DF 1600                 LD     D,0
1184: A8E1 0E30                 LD     C,CHRZER
1185: A8E3 FE64         PR10L1: CP     100              ;100 KETA
1186: A8E5 3805                 JR     C,PR10L2
1187: A8E7 0C                   INC    C
1188: A8E8 D664                 SUB    100
1189: A8EA 18F7                 JR     PR10L1
1190: A8EC CD02A9       PR10L2: CALL   PR10S
1191: A8EF FE0A         PR10L3: CP     10
1192: A8F1 3805                 JR     C,PR10L4
1193: A8F3 0C                   INC    C
1194: A8F4 D60A                 SUB    10
1195: A8F6 18F7                 JR     PR10L3
1196: A8F8 CD02A9       PR10L4: CALL   PR10S
1197: A8FB 81                   ADD    A,C
1198: A8FC CD0BA0               CALL   #PRINT
1199: A8FF D1                   POP    DE
1200: A900 C1                   POP    BC
1201: A901 C9                   RET
1202: A902              ;
1203: A902 47           PR10S:  LD     B,A
1204: A903 AF                   XOR    A
1205: A904 BA                   CP     D
1206: A905 79                   LD     A,C
1207: A906 200A                 JR     NZ,PR10S1
1208: A908 FE30                 CP     CHRZER
1209: A90A 2004                 JR     NZ,PR10S2
1210: A90C 3E20                 LD     A,CHRSPC
1211: A90E 1802                 JR     PR10S1
1212: A910 1601         PR10S2: LD     D,1
1213: A912 CD0BA0       PR10S1: CALL   #PRINT
1214: A915 78                   LD     A,B
1215: A916 0E30                 LD     C,CHRZER
1216: A918 C9                   RET
1217: A919              ;
1218: A919              ; DATA INPUT
1219: A919              ;  INPUT DATA B,(CURXS),(CURYS)
1220: A919              ;  BREAK  ALL
1221: A919              ;
1222: A919 3A5EAC       DATINP: LD     A,(CURXS)        ;CURSOR X SET
1223: A91C 5F                   LD     E,A
1224: A91D 87                   ADD    A,A
1225: A91E 87                   ADD    A,A
1226: A91F 83                   ADD    A,E
1227: A920 C60B                 ADD    A,11
1228: A922 DD7700               LD     (IX+0),A
1229: A925 3A5FAC               LD     A,(CURYS)        ;CURSOR Y SET
1230: A928 C603                 ADD    A,3
1231: A92A FE0E                 CP     14
1232: A92C 3801                 JR     C,DATIL1
1233: A92E 3C                   INC    A
1234: A92F DD7701       DATIL1: LD     (IX+1),A
1235: A932 2170AC               LD     HL,DATDAT        ;MASK DATA PUT
1236: A935 3A5FAC               LD     A,(CURYS)
1237: A938 1600         DATILA: LD     D,0
1238: A93A 5F                   LD     E,A
1239: A93B 87                   ADD    A,A
1240: A93C 83                   ADD    A,E
1241: A93D 5F                   LD     E,A
1242: A93E 19                   ADD    HL,DE
1243: A93F 7E                   LD     A,(HL)
1244: A940 4F                   LD     C,A
1245: A941 A0                   AND    B
1246: A942 47                   LD     B,A
1247: A943 C5                   PUSH   BC
1248: A944 CB39         DATIL9: SRL    C
1249: A946 3804                 JR     C,DATIL8
1250: A948 CB3F                 SRL    A
1251: A94A 18F8                 JR     DATIL9
1252: A94C C1           DATIL8: POP    BC
1253: A94D CDDDA8               CALL   PR10
1254: A950 23                   INC    HL               ;OFFSET DATA PUT
1255: A951 5E                   LD     E,(HL)           ;DATA AREA SANSHUTU 1
1256: A952 E5                   PUSH   HL
1257: A953 2A61AC               LD     HL,(DATADR)
1258: A956 19                   ADD    HL,DE
1259: A957 3A5EAC               LD     A,(CURXS)        ;DATA AREA SANSHUTU 2
1260: A95A CB3F                 SRL    A
1261: A95C 3002                 JR     NC,DATIL2
1262: A95E CBCF                 SET    1,A
1263: A960 3260AC       DATIL2: LD     (CURXSS),A
1264: A963 5F                   LD     E,A
1265: A964 19                   ADD    HL,DE
1266: A965 79                   LD     A,C              ;MASK DATA CPL
1267: A966 2F                   CPL
1268: A967 A6                   AND    (HL)
1269: A968 B0                   OR     B
1270: A969 77                   LD     (HL),A           ;DATA AREA WRITE
1271: A96A E1                   POP    HL
1272: A96B 23                   INC    HL
1273: A96C 5F                   LD     E,A
1274: A96D 56                   LD     D,(HL)           ;SUB ADR READ
1275: A96E 3E03                 LD     A,3              ;LFO DATA?
1276: A970 BA                   CP     D
1277: A971 384F                 JR     C,DATIL3
1278: A973 280F                 JR     Z,DATIL4         ;LFO DATA SHORI
1279: A975 3D                   DEC    A
1280: A976 BA                   CP     D
1281: A977 2812                 JR     Z,DATIL5
1282: A979 3D                   DEC    A
1283: A97A BA                   CP     D
1284: A97B 2845                 JR     Z,DATIL3
1285: A97D 7B                   LD     A,E
1286: A97E 214FAC               LD     HL,FVFLG         ;F.NO OR VR F
1287: A981 C3BBA9               JP     DATILC
1288: A984 7B           DATIL4: LD     A,E              ;STEP
1289: A985 2131AC               LD     HL,STEP
1290: A988 C3BBA9               JP     DATILC
1291: A98B 7B           DATIL5: LD     A,E              ;DEEP
1292: A98C 212BAC               LD     HL,DEEP
1293: A98F CDBBA9               CALL   DATILC
1294: A992 2A5EAC               LD     HL,(CURXS)
1295: A995 E5                   PUSH   HL
1296: A996 2A5FAC               LD     HL,(CURYS)
1297: A999 E5                   PUSH   HL
1298: A99A 3E03                 LD     A,3
1299: A99C 325FAC               LD     (CURYS),A
1300: A99F AF                   XOR    A
1301: A9A0 325EAC       DATILB: LD     (CURXS),A
1302: A9A3 F5                   PUSH   AF
1303: A9A4 CD21AA               CALL   DATOUT
1304: A9A7 47                   LD     B,A
1305: A9A8 CD19A9               CALL   DATINP
1306: A9AB F1                   POP    AF
1307: A9AC 3C                   INC    A
1308: A9AD 0604                 LD     B,4
1309: A9AF BF                   CP     A
1310: A9B0 20EE                 JR     NZ,DATILB
1311: A9B2 E1                   POP    HL
1312: A9B3 225FAC               LD     (CURYS),HL
1313: A9B6 E1                   POP    HL
1314: A9B7 225EAC               LD     (CURXS),HL
1315: A9BA C9                   RET
1316: A9BB 0606         DATILC: LD     B,6
1317: A9BD 77           DATILD: LD     (HL),A
1318: A9BE 23                   INC    HL
1319: A9BF 10FC                 DJNZ   DATILD
1320: A9C1 C9                   RET
1321: A9C2              ;
1322: A9C2 3A60AC       DATIL3: LD     A,(CURXSS)       ;SUB ADR SANSHUTU
1323: A9C5 87                   ADD    A,A
1324: A9C6 87                   ADD    A,A
1325: A9C7 82                   ADD    A,D
1326: A9C8 57                   LD     D,A
1327: A9C9 2603                 LD     H,3
1328: A9CB F3                   DI
1329: A9CC 010000       DATILF: LD     BC,OPN#0
1330: A9CF 7A                   LD     A,D
1331: A9D0 ED79                 OUT    (C),A
1332: A9D2 00                   NOP
1333: A9D3 03                   INC    BC
1334: A9D4 7B                   LD     A,E
1335: A9D5 00                   NOP
1336: A9D6 00                   NOP
1337: A9D7 ED79                 OUT    (C),A
1338: A9D9 00                   NOP
1339: A9DA 010200               LD     BC,OPN#1
1340: A9DD 7A                   LD     A,D
1341: A9DE ED79                 OUT    (C),A
1342: A9E0 00                   NOP
1343: A9E1 03                   INC    BC
1344: A9E2 7B                   LD     A,E
1345: A9E3 00                   NOP
1346: A9E4 00                   NOP
1347: A9E5 ED79                 OUT    (C),A
1348: A9E7 00                   NOP
1349: A9E8 14                   INC    D
1350: A9E9 00                   NOP
1351: A9EA 00                   NOP
1352: A9EB 00                   NOP
1353: A9EC 25                   DEC    H
1354: A9ED 20DD                 JR     NZ,DATILF
1355: A9EF FB                   EI
1356: A9F0 3A5FAC               LD     A,(CURYS)
1357: A9F3 FE03                 CP     3
1358: A9F5 2805                 JR     Z,DATIL6         ;TO TOTAL LEVEL
1359: A9F7 FE00                 CP     0
1360: A9F9 2815                 JR     Z,DATIL7         ;TO CONNECT
1361: A9FB C9                   RET
1362: A9FC 3A60AC       DATIL6: LD     A,(CURXSS)       ;TOTAL LEVEL
1363: A9FF 4F                   LD     C,A
1364: AA00 0600                 LD     B,0
1365: AA02 2137AC               LD     HL,TLEVEL
1366: AA05 09                   ADD    HL,BC
1367: AA06 0606                 LD     B,6
1368: AA08 73           DATILE: LD     (HL),E
1369: AA09 23                   INC    HL
1370: AA0A 23                   INC    HL
1371: AA0B 23                   INC    HL
1372: AA0C 23                   INC    HL
1373: AA0D 10F9                 DJNZ   DATILE
1374: AA0F C9                   RET
1375: AA10 2168AC       DATIL7: LD     HL,DATCON        ;CONNECT OP NO DATA
1376: AA13 1600                 LD     D,0
1377: AA15 3E07                 LD     A,7
1378: AA17 A3                   AND    E
1379: AA18 5F                   LD     E,A
1380: AA19 19                   ADD    HL,DE
1381: AA1A 7E                   LD     A,(HL)
1382: AA1B 2155AC               LD     HL,LFOCON
1383: AA1E C3BBA9               JP     DATILC
1384: AA21              ;
1385: AA21              ; DATA OUT
1386: AA21              ;   OUTPUT DATA ACC
1387: AA21              ;   BREAK  ALL
1388: AA21              ;
1389: AA21 2170AC       DATOUT: LD     HL,DATDAT
1390: AA24 3A5FAC               LD     A,(CURYS)
```

▶ぼくは通学中に「ミラノX1」にあやうくひかれそうになったことがあるのでした。
小林　敦（16）長野県

```
1391: AA27 1600          DATOL2: LD      D,0
1392: AA29 5F                    LD      E,A
1393: AA2A 87                    ADD     A,A
1394: AA2B 83                    ADD     A,E
1395: AA2C 5F                    LD      E,A
1396: AA2D 19                    ADD     HL,DE
1397: AA2E 46                    LD      B,(HL)
1398: AA2F 23                    INC     HL
1399: AA30 5E                    LD      E,(HL)
1400: AA31 2A61AC                LD      HL,(DATADR)
1401: AA34 19                    ADD     HL,DE
1402: AA35 3A5EAC                LD      A,(CURXS)
1403: AA38 CB3F                  SRL     A
1404: AA3A 3002                  JR      NC,DATOL1
1405: AA3C CBCF                  SET     1,A
1406: AA3E 5F            DATOL1: LD      E,A
1407: AA3F 19                    ADD     HL,DE
1408: AA40 78                    LD      A,B
1409: AA41 A6                    AND     (HL)
1410: AA42 C9                    RET
1411: AA43               ;
1412: AA43               ; LFO
1413: AA43               ;
1414: AA43               LFO:    ENT
1415: AA43 08                    EX      AF,AF'
1416: AA44 D9                    EXX
1417: AA45 3E01                  LD      A,1
1418: AA47 325CAC                LD      (INTFA),A
1419: AA4A 010200                LD      BC,OPN#1         ;TIMER BF RESET
1420: AA4D 3E27                  LD      A,CTSADR
1421: AA4F 0000                  DEFW    0000H            ;NOP*6
1422: AA51 0000                  DEFW    0000H
1423: AA53 0000                  DEFW    0000H
1424: AA55 ED79                  OUT     (C),A
1425: AA57 00                    NOP
1426: AA58 03                    INC     BC
1427: AA59 3A06AC                LD      A,(TCC)
1428: AA5C 00                    NOP
1429: AA5D ED79                  OUT     (C),A
1430: AA5F 00                    NOP
1431: AA60 0E00                  LD      C,0              ;COUNTER (BC REG)
1432: AA62 0600          LFOLOP: LD      B,0
1433: AA64 212BAC                LD      HL,DEEP          ;DEEP=0?(LFO OFF?)
1434: AA67 09                    ADD     HL,BC
1435: AA68 56                    LD      D,(HL)
1436: AA69 AF                    XOR     A
1437: AA6A BE                    CP      (HL)
1438: AA6B CA60AB                JP      Z,LFONEX
1439: AA6E 2119AC                LD      HL,STIMES        ;START TIME=0?
1440: AA71 09                    ADD     HL,BC
1441: AA72 BE                    CP      (HL)
1442: AA73 2804                  JR      Z,LFOSTA
1443: AA75 35                    DEC     (HL)             ;START TIME DEC
1444: AA76 C360AB                JP      LFONEX
1445: AA79 2125AC        LFOSTA: LD      HL,LFOPS         ;STTEP=0?(=0:LFO)
1446: AA7C 09                    ADD     HL,BC
1447: AA7D BE                    CP      (HL)
1448: AA7E 2804                  JR      Z,LFOLB1
1449: AA80 35                    DEC     (HL)             ;STEP DEC
1450: AA81 C360AB                JP      LFONEX
1451: AA84 2131AC        LFOLB1: LD      HL,STEP          ;LFO STEP RESET
1452: AA87 09                    ADD     HL,BC
1453: AA88 7E                    LD      A,(HL)
1454: AA89 2125AC                LD      HL,LFOPS
1455: AA8C 09                    ADD     HL,BC
1456: AA8D 77                    LD      (HL),A
1457: AA8E 2113AC                LD      HL,DEEPS         ;DEEP=DEEPS?
1458: AA91 09                    ADD     HL,BC
1459: AA92 7E                    LD      A,(HL)
1460: AA93 BA                    CP      D
1461: AA94 2805                  JR      Z,LFOUDH
1462: AA96 82                    ADD     A,D              ;-DEEPS=DEEP?
1463: AA97 2802                  JR      Z,LFOUDH
1464: AA99 1807                  JR      LFOLB3
1465: AA9B 211FAC        LFOUDH: LD      HL,LFOUD         ;LFO UP DOWN CH
1466: AA9E 09                    ADD     HL,BC
1467: AA9F 7E                    LD      A,(HL)
1468: AAA0 2F                    CPL
1469: AAA1 77                    LD      (HL),A           ;DEEPS=DEEPS+LFOUD
1470: AAA2 211FAC        LFOLB3: LD      HL,LFOUD
1471: AAA5 09                    ADD     HL,BC
1472: AAA6 7E                    LD      A,(HL)
1473: AAA7 2113AC                LD      HL,DEEPS
1474: AAAA 09                    ADD     HL,BC
1475: AAAB 86                    ADD     A,(HL)
1476: AAAC 77                    LD      (HL),A
1477: AAAD 5F                    LD      E,A              ;LFOWK(E)=DEEPS
1478: AAAE 325BAC                LD      (LFOWK),A
1479: AAB1 214FAC                LD      HL,FVFLG         ;LFO F.NO OR T.L.?
1480: AAB4 09                    ADD     HL,BC
1481: AAB5 AF                    XOR     A
1482: AAB6 BE                    CP      (HL)
1483: AAB7 CA07AB                JP      Z,LFOLBF
1484: AABA
1485: AABA C5                    PUSH    BC               ;T.LEVEL M
1486: AABB 2155AC                LD      HL,LFOCON
1487: AABE 09                    ADD     HL,BC
1488: AABF EB                    EX      DE,HL
1489: AAC0 CB21                  SLA     C
1490: AAC2 CB21                  SLA     C
1491: AAC4 2137AC                LD      HL,TLEVEL
1492: AAC7 09                    ADD     HL,BC
1493: AAC8 CB39                  SRL     C
1494: AACA CB39                  SRL     C
1495: AACC CB39                  SRL     C
1496: AACE 3808                  JR      C,LFOL#1
1497: AAD0 3E40                  LD      A,TLSADR         ;OPN#0
1498: AAD2 81                    ADD     A,C
1499: AAD3 010000                LD      BC,OPN#0
1500: AAD6 1806                  JR      LFOLB4
1501: AAD8 3E40          LFOL#1: LD      A,TLSADR
1502: AADA 81                    ADD     A,C
1503: AADB 010200                LD      BC,OPN#1
1504: AADE F5            LFOLB4: PUSH    AF
1505: AADF 1A                    LD      A,(DE)
1506: AAE0 57                    LD      D,A
1507: AAE1 F1                    POP     AF
1508: AAE2 1E04                  LD      E,4              ;OP LOOP
1509: AAE4 CB3A          LFOWLV: SRL     D
1510: AAE6 3015                  JR      NC,LFOLB5
1511: AAE8 0000                  DEFW    0000H            ;NOP*7
1512: AAEA 0000                  DEFW    0000H
1513: AAEC 0000                  DEFW    0000H
1514: AAEE 00                    DEFB    00H
1515: AAEF ED79                  OUT     (C),A
1516: AAF1 00                    NOP
1517: AAF2 0C                    INC     C
1518: AAF3 F5                    PUSH    AF
1519: AAF4 3A5BAC                LD      A,(LFOWK)
1520: AAF7 86                    ADD     A,(HL)
1521: AAF8 ED79                  OUT     (C),A
1522: AAFA 00                    NOP
1523: AAFB 0D                    DEC     C
1524: AAFC F1                    POP     AF
1525: AAFD 23            LFOLB5: INC     HL
1526: AAFE C604                  ADD     A,SADRSD
1527: AB00 1D                    DEC     E
1528: AB01 20E1                  JR      NZ,LFOWLV
1529: AB03 C1                    POP     BC
1530: AB04 C360AB                JP      LFONEX
1531: AB07               ;
1532: AB07 C5            LFOLBF: PUSH    BC               ;F.NO.M
1533: AB08 CB21                  SLA     C
1534: AB0A 2107AC                LD      HL,NOTE          ;DEEP+
1535: AB0D 09                    ADD     HL,BC
1536: AB0E CB39                  SRL     C
1537: AB10 7E                    LD      A,(HL)
1538: AB11 23                    INC     HL
1539: AB12 56                    LD      D,(HL)
1540: AB13 83                    ADD     A,E
1541: AB14 3807                  JR      C,LFOUPC
1542: AB16 CB7B                  BIT     7,E              ;DOWN BW
1543: AB18 2808                  JR      Z,LFOLB7
1544: AB1A 15                    DEC     D
1545: AB1B 1805                  JR      LFOLB7
1546: AB1D CB7B          LFOUPC: BIT     7,E              ;UP CY
1547: AB1F 2001                  JR      NZ,LFOLB7
1548: AB21 14                    INC     D
1549: AB22 5F            LFOLB7: LD      E,A
1550: AB23 79                    LD      A,C
1551: AB24 CB3F                  SRL     A
1552: AB26 C6A4                  ADD     A,FNHSAD         ;F.NO.SADR HI
1553: AB28 CB41                  BIT     0,C
1554: AB2A 2805                  JR      Z,LFOLB8
1555: AB2C 010200                LD      BC,OPN#1
1556: AB2F 1803                  JR      LFOLB9
1557: AB31 010000        LFOLB8: LD      BC,OPN#0
1558: AB34 ED79          LFOLB9: OUT     (C),A
1559: AB36 00                    NOP
1560: AB37 0C                    INC     C
1561: AB38 0000                  DEFW    0000H            ;NOP*4
1562: AB3A 0000                  DEFW    0000H
1563: AB3C ED51                  OUT     (C),D
1564: AB3E 00                    NOP
1565: AB3F 0D                    DEC     C
1566: AB40 D604                  SUB     SADRSD           ;OPN F.NO.SADR LOW
1567: AB42 0000                  DEFW    0000H            ;NOP*18
1568: AB44 0000                  DEFW    0000H
1569: AB46 0000                  DEFW    0000H
1570: AB48 0000                  DEFW    0000H
1571: AB4A 0000                  DEFW    0000H
1572: AB4C 0000                  DEFW    0000H
1573: AB4E 0000                  DEFW    0000H
1574: AB50 0000                  DEFW    0000H
1575: AB52 0000                  DEFW    0000H
1576: AB54 ED79                  OUT     (C),A
1577: AB56 00                    NOP
1578: AB57 0C                    INC     C
1579: AB58 0000                  DEFW    0000H            ;NOP*4
1580: AB5A 0000                  DEFW    0000H
1581: AB5C ED59                  OUT     (C),E
1582: AB5E 00                    NOP
1583: AB5F C1                    POP     BC
1584: AB60               ;
1585: AB60 0C            LFONEX: INC     C
1586: AB61 3E06                  LD      A,6
1587: AB63 B9                    CP      C
1588: AB64 C262AA                JP      NZ,LFOLOP
1589: AB67 AF                    XOR     A
1590: AB68 325CAC                LD      (INTFA),A
1591: AB6B 3A5DAC                LD      A,(INTFB)
1592: AB6E FE00                  CP      0
1593: AB70 2005                  JR      NZ,LFOLBA
1594: AB72 08                    EX      AF,AF'
1595: AB73 D9                    EXX
1596: AB74 FB                    EI
1597: AB75 ED4D                  RETI
1598: AB77 08            LFOLBA: EX      AF,AF'
1599: AB78 D9                    EXX
1600: AB79 FB                    EI
1601: AB7A C9                    RET
1602: AB7B               ;
1603: AB7B               ;LFO STARTER
1604: AB7B               ;
1605: AB7B (0025)        TAHSAD: EQU     25H              ;TIMER A HIGH SUB ADR
1606: AB7B (0024)        TALSAD: EQU     24H              ;TIMER A LOW SUB ADR
1607: AB7B (0027)        CTSADR: EQU     27H              ;CONTROL SUB ADR
1608: AB7B               ;
1609: AB7B               LFOSS:  ENT
1610: AB7B 010200                LD      BC,OPN#1
1611: AB7E 3E25                  LD      A,TAHSAD
1612: AB80 ED79                  OUT     (C),A
1613: AB82 00                    NOP
1614: AB83 03                    INC     BC
1615: AB84 AF                    XOR     A                ;TIMER A HIGHT DATA
1616: AB85 0000                  DEFW    0000H            ;NOP*2
1617: AB87 ED79                  OUT     (C),A
1618: AB89 00                    NOP
1619: AB8A 0B                    DEC     BC
1620: AB8B 3E24                  LD      A,TALSAD
1621: AB8D 0000                  DEFW    0000H            ;NOP*18
1622: AB8F 0000                  DEFW    0000H
1623: AB91 0000                  DEFW    0000H
1624: AB93 0000                  DEFW    0000H
1625: AB95 0000                  DEFW    0000H
1626: AB97 0000                  DEFW    0000H
1627: AB99 0000                  DEFW    0000H
1628: AB9B 0000                  DEFW    0000H
1629: AB9D 0000                  DEFW    0000H
1630: AB9F ED79                  OUT     (C),A
1631: ABA1 00                    NOP
1632: ABA2 03                    INC     BC
1633: ABA3 3E6E                  LD      A,110            ;TIMER A LOW DATA
1634: ABA5 00                    NOP
1635: ABA6 ED79                  OUT     (C),A
1636: ABA8 00                    NOP
1637: ABA9 0B                    DEC     BC
1638: ABAA 3E27                  LD      A,CTSADR
1639: ABAC 0000                  DEFW    0000H            ;NOP*18
1640: ABAE 0000                  DEFW    0000H
1641: ABB0 0000                  DEFW    0000H
1642: ABB2 0000                  DEFW    0000H
1643: ABB4 0000                  DEFW    0000H
1644: ABB6 0000                  DEFW    0000H
1645: ABB8 0000                  DEFW    0000H
1646: ABBA 0000                  DEFW    0000H
1647: ABBC 0000                  DEFW    0000H
1648: ABBE ED79                  OUT     (C),A
1649: ABC0 00                    NOP
1650: ABC1 03                    INC     BC
1651: ABC2 3A06AC                LD      A,(TCC)
1652: ABC5 00                    NOP
1653: ABC6 ED79                  OUT     (C),A
1654: ABC8 00                    NOP
1655: ABC9 C9                    RET
1656: ABCA               ;
1657: ABCA               ; INT JP S
1658: ABCA               ;
1659: ABCA C5            INTOPN: PUSH    BC
1660: ABCB F5                    PUSH    AF
1661: ABCC 010200                LD      BC,OPN#1
1662: ABCF 3E27                  LD      A,CTSADR
1663: ABD1 0000                  DEFW    0000H            ;NOP*8
1664: ABD3 0000                  DEFW    0000H
1665: ABD5 0000                  DEFW    0000H
1666: ABD7 0000                  DEFW    0000H
1667: ABD9 ED79                  OUT     (C),A
1668: ABDB 00                    NOP
1669: ABDC 03                    INC     BC
1670: ABDD 3A06AC                LD      A,(TCC)
```


▶Oh！MZにひとつ提案があるんですが，毎月「隠しプレゼント」というのを実施してみてはいかがでしょうか。表紙から最後のページにかけて場所を問わず，小さな文字でチョコッとプレゼントを入れておくのです。そうすればOh！MZのスミからスミまで読んでいる人にだけ幸運が舞い込み，立ち読みをして，愛読者カードを切り取って出すような極悪人をギャフンと言わせることができるでしょう。　石塚　孝幸（18）奈良県

```
1671: ABE0 ED79                 OUT    (C),A
1672: ABE2 00                   NOP
1673: ABE3 0B                   DEC    BC
1674: ABE4 0000                 DEFW   0000H              ;NOP*20
1675: ABE6 0000                 DEFW   0000H
1676: ABE8 0000                 DEFW   0000H
1677: ABEA 0000                 DEFW   0000H
1678: ABEC 0000                 DEFW   0000H
1679: ABEE 0000                 DEFW   0000H
1680: ABF0 0000                 DEFW   0000H
1681: ABF2 0000                 DEFW   0000H
1682: ABF4 0000                 DEFW   0000H
1683: ABF6 0000                 DEFW   0000H
1684: ABF8 ED78                 IN     A,(C)
1685: ABFA 2005                 JR     NZ,INOPL1
1686: ABFC F1                   POP    AF
1687: ABFD C1                   POP    BC
1688: ABFE C343AA               JP     LFO
1689: AC01 F1          INOPL1:  POP    AF
1690: AC02 C1                   POP    BC
1691: AC03 C343AA               JP     LFO
1692: AC06             ;
1693: AC06             ;
1694: AC06             ;SUB
1695: AC06             ;
1696: AC06             ;
1697: AC06             ;
1698: AC06             ;
1699: AC06             ; WORK
1700: AC06             ; LFO WORK
1701: AC06             ;
1702: AC06 35          TCC:     DEFB   35H                ;TIMER CONTR COM
1703: AC07             NOTE:    ENT
1704: AC07                      DEFS   12                 ;F.NO DATA
1705: AC13             DEEPS:   ENT
1706: AC13                      DEFS   6                  ;LFO DEEP NOW
1707: AC19             STIMES:  ENT
1708: AC19                      DEFS   6                  ;LFO START WAIT LEFT
1709: AC1F             LFOUD:   ENT
1710: AC1F                      DEFS   6                  ;1=UP FFH=DOWN
1711: AC25             LFOPS:   ENT
1712: AC25                      DEFS   6                  ;LFO STEP COWNTER
1713: AC2B             ;
1714: AC2B             DEEP:    ENT
1715: AC2B                      DEFS   6                  ;DEEP (=0 LFO OFF)
1716: AC31             STEP:    ENT
1717: AC31                      DEFS   6                  ;STEP
1718: AC37             TLEVEL:  ENT
1719: AC37                      DEFS   24                 ;T.LEVEL
1720: AC4F             FVFLG:   ENT
1721: AC4F                      DEFS   6                  ;LFO F.NO OR VR
1722: AC55             LFOCON:  ENT
1723: AC55                      DEFS   6                  ;LFO VR OP.NO.
1724: AC5B             LFOWK:   DEFS   1                  ;WORK
1725: AC5C             INTFA:   DEFS   1                  ;INTF A
1726: AC5D             INTFB:   DEFS   1                  ;INTF B:1=INT
1727: AC5E             ;
1728: AC5E             ; OTOZUKURI WORK
1729: AC5E             ;
1730: AC5E             CURXS:   DEFS   1                  ;HYOU NO CURSOR
1731: AC5F             CURYS:   DEFS   1
1732: AC60             CURXSS:  DEFS   1                  ;HTOU CUR(OP DATA YO)
1733: AC61             DATADR:  DEFS   2                  ;IMA NO DATA TOP ADR
1734: AC63             NOBASU:  DEFS   1                  ;NOBASU/NOBASANAI
1735: AC64             KEYB:    DEFS   1                  ;KEY BF
1736: AC65             OCT:     DEFS   1                  ;OCTABU
1737: AC66             PARTNO:  DEFS   1                  ;PART NO.
1738: AC67             DATNO:   DEFS   1                  ;DATA NO.
1739: AC68             ;
1740: AC68             ; OTOZUDURI DATA
1741: AC68             ;
1742: AC68 08          DATCON:  DEFB   08H                ;CONN0 CR=1,MO=0
1743: AC69 08                   DEFB   08H                ;CONN1
1744: AC6A 08                   DEFB   08H                ;CONN2
1745: AC6B 08                   DEFB   08H                ;CONN3
1746: AC6C 0C                   DEFB   0CH                ;CONN4
1747: AC6D 0E                   DEFB   0EH                ;CONN5
1748: AC6E 0E                   DEFB   0EH                ;CONN6
1749: AC6F 0F                   DEFB   0FH                ;CONN7
1750: AC70 07          DATDAT:  DEFB   07H                ;CONNECT
1751: AC71 23                   DEFB   35
1752: AC72 B0                   DEFB   B0H
1753: AC73 38                   DEFB   38H                ;F.BACK
1754: AC74 23                   DEFB   35
1755: AC75 B0                   DEFB   B0H
1756: AC76 0F                   DEFB   0FH                ;MULTIPLE
1757: AC77 0B                   DEFB   11
1758: AC78 30                   DEFB   30H
1759: AC79 7F                   DEFB   7FH                ;T.LEVEL
1760: AC7A 0F                   DEFB   15
1761: AC7B 40                   DEFB   40H
1762: AC7C 1F                   DEFB   1FH                ;A.RATE
1763: AC7D 13                   DEFB   19
1764: AC7E 50                   DEFB   50H
1765: AC7F 1F                   DEFB   1FH                ;D.RATE
1766: AC80 17                   DEFB   23
1767: AC81 60                   DEFB   60H
1768: AC82 1F                   DEFB   1FH                ;S.RATE
1769: AC83 1B                   DEFB   27
1770: AC84 70                   DEFB   70H
1771: AC85 F0                   DEFB   F0H                ;S.LEVEL
1772: AC86 1F                   DEFB   31
1773: AC87 80                   DEFB   80H
1774: AC88 0F                   DEFB   0FH                ;R.RATE
1775: AC89 1F                   DEFB   31
1776: AC8A 80                   DEFB   80H
1777: AC8B 70                   DEFB   70H                ;DETUNE
1778: AC8C 0B                   DEFB   11
1779: AC8D 30                   DEFB   30H
1780: AC8E C0                   DEFB   C0H                ;K.SCALE
1781: AC8F 13                   DEFB   19
1782: AC90 50                   DEFB   50H
1783: AC91 01                   DEFB   01H                ;LFO FN/TL
1784: AC92 24                   DEFB   36
1785: AC93 00                   DEFB   0
1786: AC94 FF                   DEFB   FFH                ;LFO S.WAIT
1787: AC95 25                   DEFB   37
1788: AC96 01                   DEFB   1
1789: AC97 7F                   DEFB   7FH                ;LFO DEEP
1790: AC98 26                   DEFB   38
1791: AC99 02                   DEFB   2
1792: AC9A FF                   DEFB   FFH                ;LFO STEP
1793: AC9B 27                   DEFB   39
1794: AC9C 03                   DEFB   3
1795: AC9D                ;INKEY DATA
1796: AC9D 31                   DEFB   31H                ;1 0
1797: AC9E 51                   DEFB   51H                ;Q 1
1798: AC9F 57                   DEFB   57H                ;W 2
1799: ACA0 33                   DEFB   33H                ;3 3
1800: ACA1 45                   DEFB   45H                ;E 4
1801: ACA2 34                   DEFB   34H                ;4 5
1802: ACA3 52                   DEFB   52H                ;R 6
1803: ACA4 54                   DEFB   54H                ;T 7
1804: ACA5 36                   DEFB   36H                ;6 8
1805: ACA6 59                   DEFB   59H                ;Y 9
1806: ACA7 37                   DEFB   37H                ;7 10
1807: ACA8 55                   DEFB   55H                ;U 11
1808: ACA9 38                   DEFB   38H                ;8 12
1809: ACAA 49                   DEFB   49H                ;I 13
1810: ACAB 4F                   DEFB   4FH                ;O 14
1811: ACAC 30                   DEFB   30H                ;0 15
1812: ACAD 50                   DEFB   50H                ;P 16
1813: ACAE 41                   DEFB   41H                ;A 17
1814: ACAF 5A                   DEFB   5AH                ;Z 18
1815: ACB0 53                   DEFB   53H                ;S 19
1816: ACB1 58                   DEFB   58H                ;X 20
1817: ACB2 44                   DEFB   44H                ;D 21
1818: ACB3 43                   DEFB   43H                ;C 22
1819: ACB4 4D                   DEFB   4DH                ;M 23
1820: ACB5 2C                   DEFB   2CH                ;, 24
1821: ACB6 2E                   DEFB   2EH                ;. 25
1822: ACB7 4B                   DEFB   4BH                ;K 26
1823: ACB8 0D                   DEFB   0DH                ;CR 27
1824: ACB9 20                   DEFB   20H                ;SPC 28
1825: ACBA 4E                   DEFB   4EH                ;N 29
1826: ACBB 46                   DEFB   46H                ;F 30
1827: ACBC 56          KEYDAT:  DEFB   56H                ;V 31
1828: ACBD 2602        NOTDAT:  DEFW   0226H              ;3 #A
1829: ACBF 4702                 DEFW   0247H              ;3  B
1830: ACC1 6902                 DEFW   0269H              ;4  C
1831: ACC3 8E02                 DEFW   028EH              ;4 #C
1832: ACC5 B502                 DEFW   02B5H              ;4  D
1833: ACC7 DE02                 DEFW   02DEH              ;4 #D
1834: ACC9 0A03                 DEFW   030AH              ;4  E
1835: ACCB 3803                 DEFW   0338H              ;4  F
1836: ACCD 6903                 DEFW   0369H              ;4 #F
1837: ACCF 9D03                 DEFW   039DH              ;4  G
1838: ACD1 D403                 DEFW   03D4H              ;4 #G
1839: ACD3 0E04                 DEFW   040EH              ;4  A
1840: ACD5 4C04                 DEFW   044CH              ;4 #A
1841: ACD7 8D04                 DEFW   048DH              ;4  B
1842: ACD9 D304                 DEFW   04D3H              ;5  C
1843: ACDB 1C05                 DEFW   051CH              ;5 #C
1844: ACDD 6A05                 DEFW   056AH              ;5  D
1845: ACDF 4E4F        M1:      DEFM   'NO'
1846: ACE1 0D                   DEFB   0DH
1847: ACE2 20202020    M2:      DEFM   '              1     2     3     4'
    : ACE6 20202020
    : ACEA 20202020
    : ACEE 31202020
    : ACF2 20322020
    : ACF6 20203320
    : ACFA 20202034
1848: ACFE 0D                   DEFB   0DH
1849: ACFF 20434F4E    M3:      DEFM   ' CONNECT         ---   ---   ---'
    : AD03 4E454354
    : AD07 20202020
    : AD0B 20202020
    : AD0F 2D2D2D20
    : AD13 202D2D2D
    : AD17 20202D2D
    : AD1B 2D
1850: AD1C 0D                   DEFB   0DH
1851: AD1D 20462E42    M4:      DEFM   ' F.BACK          ---   ---   ---
    : AD21 41434B20
    : AD25 20202020
    : AD29 20202020
    : AD2D 2D2D2D20
    : AD31 202D2D2D
    : AD35 20202D2D
    : AD39 2D
1852: AD3A 0D                   DEFB   0DH
1853: AD3B 204D554C    M5:      DEFM   ' MULTIPLE'
    : AD3F 5449504C
    : AD43 45
1854: AD44 0D                   DEFB   0DH
1855: AD45 20542E4C    M6:      DEFM   ' T.LEVEL'
    : AD49 4556454C
1856: AD4D 0D                   DEFB   0DH
1857: AD4E 20412E52    M7:      DEFM   ' A.RATE'
    : AD52 415445
1858: AD55 0D                   DEFB   0DH
1859: AD56 20442E52    M8:      DEFM   ' D.RATE'
    : AD5A 415445
1860: AD5D 0D                   DEFB   0DH
1861: AD5E 20532E52    M9:      DEFM   ' S.RATE'
    : AD62 415445
1862: AD65 0D                   DEFB   0DH
1863: AD66 20532E4C    M10:     DEFM   ' S.LEVEL'
    : AD6A 4556454C
1864: AD6E 0D                   DEFB   0DH
1865: AD6F 20522E52    M11:     DEFM   ' R.RATE'
    : AD73 415445
1866: AD76 0D                   DEFB   0DH
1867: AD77 20444554    M12:     DEFM   ' DETUNE'
    : AD7B 554E45
1868: AD7E 0D                   DEFB   0DH
1869: AD7F 204B2E53    M13:     DEFM   ' K.SCALE'
    : AD83 43414C45
1870: AD87 0D                   DEFB   0DH
1871: AD88 4C464F      M14:     DEFM   'LFO'
1872: AD8B 0D                   DEFB   0DH
1873: AD8C 20464E2F    M15:     DEFM   ' FN/TL'
    : AD90 544C
1874: AD92 0D                   DEFB   0DH
1875: AD93 20532E57    M16:     DEFM   ' S.WAIT'
    : AD97 414954
1876: AD9A 0D                   DEFB   0DH
1877: AD9B 20444545    M17:     DEFM   ' DEEP'
    : AD9F 50
1878: ADA0 0D                   DEFB   0DH
1879: ADA1 20535445    M18:     DEFM   ' STEP'
    : ADA5 50
1880: ADA6 0D                   DEFB   0DH
1881: ADA7 204F4354    M19:     DEFM   ' OCT'
1882: ADAB 0D                   DEFB   0DH
1883: ADAC 4E414D45    M21:     DEFM   'NAME'
1884: ADB0 0D                   DEFB   0DH
1885: ADB1 4E454952    M22:     DEFM   'NEIRO'
    : ADB5 4F
1886: ADB6 0D                   DEFB   0DH
1887: ADB7 4C4F4144    M23:     DEFM   'LOAD'
1888: ADBB 0D                   DEFB   0DH
1889: ADBC 53415645    M24:     DEFM   'SAVE'
1890: ADC0 0D                   DEFB   0DH
1891: ADC1 434F5059    M25:     DEFM   'COPY'
1892: ADC5 0D                   DEFB   0DH
1893: ADC6 45584954    M30:     DEFM   'EXIT'
1894: ADCA 0D                   DEFB   0DH
1895: ADCB 3E3D        M31:     DEFM   '>='
1896: ADCD 0D                   DEFB   0DH
1897: ADCE 20202020    M26:     DEFM   '     '
    : ADD2 20
1898: ADD3 20202020    M27:     DEFM   '
    : ADD7 20202020
    : ADDB 20202020
    : ADDF 20202020
    : ADE3 202020
1899: ADE6 0D                   DEFB   0DH
1900: ADE7 46494C45    M28:     DEFM   'FILE'
1901: ADEB 0D                   DEFB   0DH
1902: ADEC 3D3D3D3D    M29:     DEFM   '======='
    : ADF0 3D3D3D
1903: ADF3 0D                   DEFB   0DH
1904: ADF4 FB          EIGKEY:  EI
1905: ADF5 C3D01F               JP     1FD0H              ;GET KEY
1906: ADF8                      END
```

▶X1FのあとにでたからX1turboII＝X1G。ブラック仕様は限定だからlimited。したがってX1turboII＝X1G－limited。オプションでタイヤがついたりするわけはないなー。

龍尾　謙二（19）愛知県

マシン語体操1・2・3　Exercise 2

計算の極意は旗にあり

Izumi Daisuke
泉 大介

前回はLD, CALLの2つの命令の使用法を解説しました。Z80はレジスタという変数のようなものを持っていて，これに値を代入するのがLDの役目です。そしてBASICのGOSUBにあたるもの，これがCALLです。「CALL 1FC4H」とやれば，1FC4H番地からはじまるマシン語のサブルーチンを呼び出します。S-OSの1FC4H番地にはBEEP音を鳴らすサブルーチンが入っていますから，「CALL 1FC4H」を実行するとピッと音が鳴るという寸法です。

まずはフラグのお話

さて，今回はZ80でできる演算を解説します。演算は大きく2つに分けることができ，それぞれ算術演算，論理演算と呼ばれています。このうち論理演算については次号で扱うことにして，今回は算術演算を中心にフラグの話と合わせて講義を進めていきましょう。

演算を修得するときに決して避けて通ることができないものがあります。それはフラグです。フラグというのは旗という意味で，処理の流れを制御するという大切な役目を負っています。いわばマシン語プログラミングのキモにあたるもので，フラグの使い方がわかればマシン語の峠は越えたといえるほどです。「フラグを制する者はマシン語を制する」というわけです。

レジスタはA～Lまでありますが，それぞれのレジスタに入れることのできる数はいくつからいくつまでだったか覚えてますか。そう，00H～FFHですね。

では，FFHに01H を加えるとどうなると思います？　本来ならFFH+01H=100Hとなるところなのですが，実はFFHに01Hを加えると00Hになってしまうのです。これはどうしたことか，といいますと，ご存じのようにレジスタには16進で2桁の数しか入れることができません。そこで，FFH+01H=100Hという答えの下2桁をとってきて00Hとなるのです。このとき，繰り上がりがあったよ(オーバーフロー)という意味で旗が立ちます(セットされる)。これをキャリフラグと呼びます。繰り上がりがなかった場合，キャリフラグは立ちません(リセットされる)。反対に00Hから01Hを引くとどうなるか。今度はFFHになってしまいます。00Hから01Hを引く。引こうにも0から1は引けませんね，そこで100H－01H=FFHとして，1借りましたよ（アンダーフロー）というしるしに旗を立てます。これもキャリフラグです。

つまりキャリフラグは，実際には存在しない仮想の3桁目に繰り上がりがあったとき，仮想の3桁目から1借りてきたときに立つといえます。

もうおわかりかと思いますが，このようにレジスタの値を増加させていくと，…, FEH, FFH, 00H, 01H,…と変わっていきます。減少させていくと，…, 00H, FFH,… のようになります。

演算の結果，レジスタの値が00Hになることがありますね。このときにも旗が立ちます。これがゼロフラグです。結果が00H にならないときはゼロフラグはリセットされます。

レジスタの値を増加・減少させていくときの様子をまとめて図1に示します。いつ，どんなフラグが立つのか，ということに注意して見てください。

前回のお話をよーく読んでくださった方は，ここまできて「ねーねーじゃあHLとかBCはどうなの？」という疑問をお持ちになったでしょう。う～んあなたは優秀ですね。レジスタペア，HL, DE, BCは0000H～FFFFHの値しかとれないことはお話しました。ということは，やはりFFFFHの次は0000Hなんですよ。ところが！フラグは尋常ではない立ち方をします。命令とのかね合いもありますので，詳細は後述することにしましょう。

足し算，引き算

Z80の算術演算には次のようなものがあります。

・加算　ADD, ADC

図1　値の増減とフラグの変化

```
増加……→FEH→FFH→00H→01H→…
         NC,NZ  NC,NZ  C,Z   NC,NZ

減少……→01H→00H→FFH→FEH→…
         NC,NZ  NC,Z  C,NZ  NC,NZ

  C, キャリ      キャリフラグが立っている
 NC, ノンキャリ  キャリフラグが立ってない
  Z, ゼロ        ゼロフラグが立っている
 NZ, ノンゼロ    ゼロフラグが立ってない
```

・減算　SUB, SBC
・増加　INC
・減少　DEC

驚いたことに乗算，除算は用意されていません。乗算，除算をやりたいときには，自分でプログラムを組んでやらなければならないのです。この方法はあとで述べるとして，各命令の説明をしていきましょう。

加算 (1)

〈文法1〉

```
ADD    A, m     mはA～L，2桁の16進数，(HL)
ADD    HL, rp   rpはBC, DE, HL
```

ADDはアドまたはアッドと読み，加えるという意味です。たとえば，

　ADD　A, 20H

は，A＝A＋20Hとなります。加算した結果は必ずAまたはHLに入ります。AとHLが特別に扱われているのがわかりますか？

　ADD　B, D
　ADD　DE, BC

という命令は存在しないのです。ですから「ADD　B, D」をやりたければ次のような手を使わなければなりません。

　LD　A, B
　ADD　A, D
　LD　B, A

これで「ADD　B, D」と同じ結果を得ることができます。

例1　足し算 (ADD)

```
0000                1 ;    SAMPLE-1
0000                2 ;
0000                3
8000                4              ORG      8000H
8000                5              ;
8000                6 PRINT:       EQU      1FF4H
8000                7 PRNTS:       EQU      1FF1H
8000                8 PRTHX:       EQU      1FC1H
8000                9              ;
8000 06 FF         10 START:       LD       B,0FFH
8002 0E 05         11              LD       C,5
8004 78            12              LD       A,B
8005 CD C1 1F      13              CALL     PRTHX
8008 CD F1 1F      14              CALL     PRNTS
800B 3E 2B         15              LD       A,'+'
800D CD F4 1F      16              CALL     PRINT
8010 CD F1 1F      17              CALL     PRNTS
8013 79            18              LD       A,C
8014 CD C1 1F      19              CALL     PRTHX
8017 CD F1 1F      20              CALL     PRNTS
801A 3E 3D         21              LD       A,'='
801C CD F4 1F      22              CALL     PRINT
801F CD F1 1F      23              CALL     PRNTS
8022 78            24              LD       A,B
8023 81            25              ADD      A,C
8024 CD C1 1F      26              CALL     PRTHX
8027 C9            27              RET
```

先ほど，レジスタの値はぐるぐる循環するものであることを紹介しました。例1はこれを実際に目で見て確かめようというものです。STARTの部分でBとCに入れている値を適当に変えて試してみてください。プログラム中でコールしているサブルーチンはS-OS中のサブルーチンです。

この中に注意すべきサブルーチンがあります。それはPRTHXです。6月号のP.39の左上に説明がありますが，その欄を右にたどってください。レジスタ破壊のところがAFとなっていますね。ということは，このサブルーチンをコールするとAの内容が変わってしまうのです。仮に

　LD　A, 1
　CALL　PRTHX

というプログラムを組むとしますね。そうすると画面に01と表示しますが，この時点でA≠1となり，次に「ADD　A, 2」とかやると悲惨な結果を招いてしまうことになります。自分でプログラムを組むときには，このレジスタ破壊にはくれぐれも注意してくださいね。

例1と同じ手法を用いて，レジスタペア（HL, DE, BC)の値も循環するものであることを確かめることができます。各自挑戦してみてください。

減算 (1)

〈文法2〉

```
SUB    m      A=A-m
```

SUBはサブまたはサブトラクトと読み，引くという意味の英語Subtract の略です。今回は「SUB　A, m」とは書いてありませんね。なぜかというと,「SUB　HL, rp」という16ビットの減算は用意されていないため,「SUB　A, m」とわざわざ書かなくても,「SUB　m」だけでわかるからなのです。

例1で15行と25行を以下のように書き換えてやれば，減算をした場合のレジスタの値の変化を見ることができます。

```
800B   3E  2D        15      LD      A,'-'
8023   91            25      SUB     C
```

加算 (2)

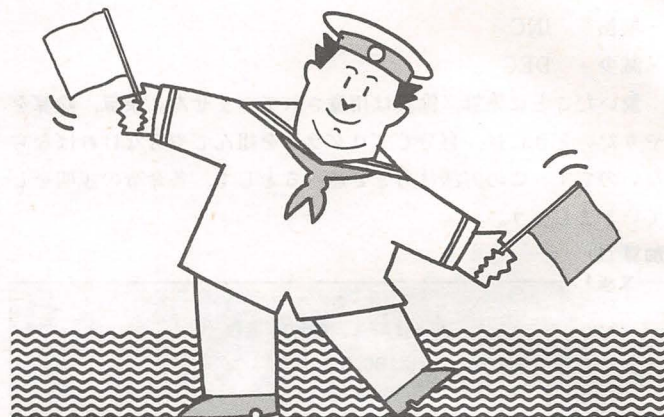
〈文法3〉

```
ADC    A, m      A=A+m+CY
ADC    HL, rp    HL=HL+rp+CY
         CYはキャリフラグが立っているとき1, そうでなければ0
```

この命令はアドキャリまたはアドウィズキャリと読みます。キャリフラグが立っているときには，通常のADDの答えにさらに1を加えるという動作をします。

例2　足し算 (ADC)

```
0000                1 ;    SAMPLE-2
0000                2 ;
0000                3
8000                4              ORG      8000H
8000                5              ;
8000                6 PRTHL:       EQU      1FBEH
8000                7 LETNL:       EQU      1FEEH
8000                8
8000 21 FF 01       9              LD       HL,01FFH
8003 3E 02         10              LD       A,2          ; A=2
8005 85            11              ADD      A,L          ; A+=FF CY=1
8006 6F            12              LD       L,A          ; L=01
8007 7C            13              LD       A,H          ; A=01
8008 CE 00         14              ADC      A,0          ; A+0+CY=2
800A 67            15              LD       H,A          ; H=2
800B CD BE 1F      16              CALL     PRTHL
800E CD EE 1F      17              CALL     LETNL
8011               18              ;
8011 21 FF 01      19              LD       HL,01FFH
8014 11 02 00      20              LD       DE,2
8017 19            21              ADD      HL,DE
8018 CD BE 1F      22              CALL     PRTHL
801B C9            23              RET
```

例2では$01FF_H+0003_H$をやって表示してみました。同じことを19行以降でもやっていますが，こっちはDEを壊してしまうというデメリットがあります。キャリ付きの加算のメリットは，ほかのレジスタの値を壊さずに加算できるだけではありません。

例3　32ビットの足し算

```
0000                 1 ;  SAMPLE-3
0000                 2 ;
0000                 3
8000                 4          ORG     8000H
8000                 5          ;
8000                 6 PRNTS:   EQU     1FF1H
8000                 7 PRINT:   EQU     1FF4H
8000                 8 PRTHL:   EQU     1FBEH
8000                 9          ;
8000 01 34 12       10          LD      BC,1234H      ; ﾀｻﾚﾙ ｶｽﾞ
8003 11 00 FE       11          LD      DE,0FE00H     ; 1234FE00 H
8006                12          ;
8006 CD 39 80       13          CALL    @PRNT
8009 3E 2B          14          LD      A,'+'
800B CD F4 1F       15          CALL    PRINT         ; PRINT "+";
800E CD F1 1F       16          CALL    PRNTS
8011                17          ;
8011 ED 53 47 80    18          LD      (WORK),DE     ;   (WORK)=E
8015                19                                ; (WORK+1)=D
8015 ED 43 49 80    20          LD      (WORK+2),BC   ; (WORK+2)=C
8019                21                                ; (WORK+3)=B
8019                22          ;
8019 01 00 00       23          LD      BC,0          ; ﾀｽｶｽﾞ
801C 11 00 02       24          LD      DE,0200H      ; 00000200 H
801F                25          ;
801F CD 39 80       26          CALL    @PRNT
8022 3E 3D          27          LD      A,'='
8024 CD F4 1F       28          CALL    PRINT         ; PRINT "=";
8027 CD F1 1F       29          CALL    PRNTS
802A                30          ;
802A 2A 47 80       31          LD      HL,(WORK)     ; HL=ﾀｻﾚﾙ DE
802D 00             32          NOP                   ; ﾅﾆﾓ ｼﾅｲ
802E 19             33          ADD     HL,DE         ; CY = 1
802F 00             34          NOP                   ; ﾅﾆﾓ ｼﾅｲ
8030 5D             35          LD      E,L
8031 54             36          LD      D,H           ; DE = HL
8032 2A 49 80       37          LD      HL,(WORK+2)   ; HL=ﾀｻﾚﾙ BC
8035 ED 4A          38          ADC     HL,BC         ; HL+BC+CY
8037 4D             39          LD      C,L
8038 44             40          LD      B,H           ; BC = HL
8039                41
8039 69             42 @PRNT:   LD      L,C
803A 60             43          LD      H,B
803B CD BE 1F       44          CALL    PRTHL         ; PRINT BC
803E 6B             45          LD      L,E
803F 62             46          LD      H,D
8040 CD BE 1F       47          CALL    PRTHL         ; PRINT DE
8043 CD F1 1F       48          CALL    PRNTS
8046 C9             49          RET
8047                50
8047 00 00          51 WORK:    DEFW    0
8049 00 00          52          DEFW    0
```

例3では32ビット長の数の足し算，$1234FE00_H+200_H$をやっています。10行でBCDEに数を代入，表示したあと，18行でWORKにとっておきます。21行で再びBCDEに数を代入。このBCDEと先にWORKにとっておいた数の加算を29行から行い，答えを表示して帰ってくるというプログラムです。プログラム中の何カ所かで「CALL　@PRNT」とやっていますね。@PRNTというラベルは例3の終わりのほうにあるサブルーチンで，これは私が作ったものです。このようにすれば，自分で作ったサブルーチンをコールすることができるのです。何度も同じ処理を書くのはバカらしい上にプログラムも大きくなってしまいますからね。

例3を見ておわかりのように,プログラム中にLD命令の何と多いことか。続いてCALL命令が多いでしょう？　実際のプログラムとはこんなもので，大半をLD，CALLが占めています。先月，しょっぱなにLDとCALLをやってしまったのは，この辺の事情があったからなのです。

この例3を少し詳しく見てみましょう。最初にBCDEに設定しているのは「足される数」です。次に@PRNTをコールしています。ここで，BCとDEに入っている値を続けて表示しますね。後ろにスペースを表示して帰ってくるようにできています。「＋」とスペースを表示したあとレジスタの内容をWORKにとっておきます。なぜ「LD (WORK+1), BC」としてないのか。先月号でお話ししましたね。思い出してください。

次に再びBCDEに値を設定しています。今度は「足す数」です。@PRNTをコールして表示したあと，「＝」をプリントしています。

さて，ここからが今回の目玉です。

「LD　HL,(WORK)」で先にメモリにとっておいたDEの値がHLに入ります。足される数の下4桁ですね。つまり$HL=FE00_H$となります。これとDEを足してますね。$HL=FE00_H+0200_H=0000_H$となり，仮想の5桁目へ繰り上がりがあったよ，とキャリフラグが立ちます(そう作ってあるのだ)。計算した結果をDEに戻して，今度は上4桁の計算に入ります。

「LD　HL,(WORK+2)」で$HL=1234_H$になります。足す数の上4桁は0000_Hですから，HLと足すと1234_H？　いえいえ，繰り上がりがあったのを忘れてはいけません。繰り上がりを計算に入れるために「ADC　HL, BC」が使ってあるのです。今キャリフラグが立ってますから，文法3でCY＝1。$HL+BC+CY=1234_H+0+1=1235_H$。この値がHLに入ります。結果をBCに戻して @PRNTをコールすると，計算結果を表示します。下4桁の計算の結果繰り上がりがないときは？　CY＝0ですからHL+BC+CY＝HL+BCですね。自分で適当な数を「足す数」や「足される数」に設定してプログラムを追ってみてください。それほど難しいことではありませんのできっと理解できるはずです。

途中2カ所NOP（ノーオペレーション，何もしない）が入れてありますが，これは文法4のためです。プログラムの流れにはまったく関係がないので，打ち込むだけ打ち込んだら，無視してください。

減算（2）

〈文法4〉

SBC	A, m	A＝A－m－CY
SBC	HL, rp	HL＝HL－rp－CY

CYはキャリフラグが立っていると1，そうでないと0

これはサブキャリ（サブウィズキャリ，サブトラクトウィズキャリ）と読みます。今度はキャリフラグが立っていると通常のSUBからさらに1を引きます。

もうお気付きの方もいるでしょう。そうです，SBCには16ビッ

トの減算もあるのです。ということは，キャリフラグが立っていないとき，「SBC　HL，BC」は「SUB　HL，BC」（こんな命令はないんですよ）と同じ働きをしてくれるということではないか！というわけで，単純にHLからBCを引きたいときには次のような手を使います。

```
OR      A
SBC     HL, BC
```

ORというのは次回説明する論理演算命令のひとつで，「OR　A」にはキャリフラグをリセットする働きがあります。この手法を使えば純粋にHL－BCを計算できるというわけです。

例3の以下の行を変更すれば，32ビットの減算プログラムのできあがりです。

```
8009  3E2D   14行   LD   A,'－'
802D  B7     30行   OR   A
802E  ED52   31行   SBC  HL, DE
             32行   ;
8035  ED42   36行   SBC  HL, BC
```

定数倍の仕方

プログラムを作っている最中に「ちょっとこの値を3倍させたいな」とか「Bを10倍しておきたい」とか思うことがあるでしょう？速い掛け算ルーチンを作るのは，実際うんざりするほど面倒なものです。また，ちょっとした掛け算のために，わざわざそんな思いをするのもバカらしいことでしょう。そこで，簡単な掛け算（定数倍）の仕方をここで練習してみることにします。

```
ADD  A, A
```

これはA＝A＋Aをやる命令でしたね。視点を変えてみると，同じものを足すのは2倍するのと同じこと，つまりA＝A×2と同じことだということに気付きます。

```
ADD     HL, HL
ADD     HL, HL
```

というようにADDを2回やると，2倍したものをさらに2倍するわけですから4倍したのと同じことになります。ADDを3回やると8倍，4回で16倍。このように考えると64倍や128倍なんてすぐにできますね。

では，3倍するときはどうしましょう。3倍には例4のような手を使います。

10行からはAを3倍，17行からはHLを3倍しています。これは，A×3＝A×2＋A，HL×3＝HL×2＋HLと考えて結構です。　この方法に習って展開すれば、ある程度の掛け算なら作ってやることができます。

A×2＝A＋A
A×3＝A×2＋A
A×4＝A×2×2
A×5＝A×2×2＋A
A×9＝A×2×2×2＋A　　　（HLも同様）

以上は，例4のように最初のAの値をほかのレジスタにコピーしておいてやることで実現できます。試しにA×9をやってみると，例5のようになります。

例4　3倍

```
0000              1 ;     SAMPLE-4
0000              2 ;
0000              3
8000              4        ORG    8000H
8000              5 PRTHX:  EQU    1FC1H
8000              6 PRTHL:  EQU    1FBEH
8000              7 LETNL:  EQU    1FEEH
8000              8
8000              9 ;A = A * 3
8000 3E 01       10        LD     A,1        ; A = 1
8002 47          11        LD     B,A        ; B = A
8003 87          12        ADD    A,A        ; A *= 2
8004 80          13        ADD    A,B        ; A *= 3
8005 CD C1 1F    14        CALL   PRTHX
8008 CD EE 1F    15        CALL   LETNL
800B             16 ;HL = HL * 3
800B 21 00 10    17        LD     HL,1000H
800E 5D          18        LD     E,L        ; DE = HL
800F 54          19        LD     D,H
8010 29          20        ADD    HL,HL      ; HL *= 2
8011 19          21        ADD    HL,DE      ; HL *= 3
8012 CD BE 1F    22        CALL   PRTHL
8015 C9          23        RET
```

例5　9倍

```
0000              1 ;     SAMPLE-5
0000              2 ;
8000              3        ORG    8000H
8000              4 PRTHX:  EQU    1FC1H
8000              5
8000              6 ;A = A * 9
8000 3E 01        7        LD     A,1        ; A = 1
8002 47           8        LD     B,A        ; B = A
8003 87           9        ADD    A,A        ; A *= 2
8004 87          10        ADD    A,A        ; A *= 4
8005 87          11        ADD    A,A        ; A *= 8
8006 80          12        ADD    A,B        ; A *= 9
8007 CD C1 1F    13        CALL   PRTHX
800A C9          14        RET
```

6倍，7倍，10倍もちょっと工夫してやればほかのレジスタを1個しか壊さずに実現してやることができます。

6倍は，　(A×2＋A)×2
7倍は，　A×2×2×2－A
10倍は，　A×2×2×2＋A×2

例6　6，7，8倍

```
0000              1 ;     SAMPLE-6
0000              2 ;
8000              3        ORG    8000H
8000              4        ;
8000              5 PRTHX:  EQU    1FC1H
8000              6 LETNL:  EQU    1FEEH
8000              7
8000              8 ;A = A * 6
8000 3E 01        9        LD     A,1        ; A = 1
8002 47          10        LD     B,A        ; B = A
8003 87          11        ADD    A,A        ; A *= 2
8004 80          12        ADD    A,B        ; A *= 3
8005 87          13        ADD    A,A        ; A *= 6
8006 CD C1 1F    14        CALL   PRTHX
8009 CD EE 1F    15        CALL   LETNL
800C             16 ;A = A * 7
800C 3E 01       17        LD     A,1        ; A = 1
800E 47          18        LD     B,A        ; B = A
800F 87          19        ADD    A,A        ; A *= 2
8010 87          20        ADD    A,A        ; A *= 4
8011 87          21        ADD    A,A        ; A *= 8
8012 90          22        SUB    B          ; A *= 7
8013 CD C1 1F    23        CALL   PRTHX
8016 CD EE 1F    24        CALL   LETNL
8019             25 ;A = A * 10
8019 3E 01       26        LD     A,1        ; A = 1
801B 87          27        ADD    A,A        ; A *= 2
801C 47          28        LD     B,A        ; B = A
801D 87          29        ADD    A,A        ; A *= 4
801E 87          30        ADD    A,A        ; A *= 8
801F 80          31        ADD    A,B        ; A *= 10
8020 CD C1 1F    32        CALL   PRTHX
8023 CD EE 1F    33        CALL   LETNL
8026             34        ;
8026 C9          35        RET
```

と考えてプログラムしてあります。2倍は簡単にできるので，これをいかにうまく使うかが鍵となります。

ここにあげた2～10倍で，定数倍についてはまずこと足りるでしょう。VRAMのアドレス計算に必要な40倍，80倍も，この方法を応用すればすぐに組めると思います。リストを見て自分で研究してみてください。

〈文法5〉

```
INC    r     r=r+1      rは，A～L，(HL)
DEC    r     r=r-1      。
INC    rp    rp=rp+1
DEC    rp    rp=rp-1
```

レジスタの値の増減

最後はレジスタなどの値を1増やしたり1減らしたりする方法です。INCはインクリメント，DECはデクリメントと読みます。

今A＝34_Hだとしますね。「INC　A」をやるとA＝35_H になります。ゼロフラグとキャリフラグは？　そう，どちらもリセットされます。B＝0のときに「DEC　B」をやるとどうなるでしょう。BはFF_Hとなってキャリフラグが立ちそうですね。

HL＝$FFFF_H$とします。「INC　HL」を行ったときのフラグの変化はどうなるでしょう。「キャリとゼロのフラグが立ちます」と答えた方は今までの話を十分に理解なさっている方です。

しかしながら，ここでちょっとやっかいな問題を持ち込まなければなりません。冒頭で，「レジスタペアに対する演算ではフラグが尋常ではない立ち方をする」と述べたことを思い出してください。実は「INC　HL」をやってもフラグは変化しないのです。それだけではなく，先の「DEC　B」でもキャリフラグは変化してくれません。

フラグが変化しないというのは「INC　HL」の前のフラグの状態がそのまま保存されるということです。キャリが立っていたら「INC　HL」実行後もキャリは立ったまま，ゼロフラグが立っていたら立ったまま，ということになります。今回解説した命令とフラグの変化の対応表を表1にまとめておきます。

```
LD     HL，8000H
LD     DE，8000H
ADD    HL，DE
```

の結果，HL＝0000_Hとなりキャリフラグが立ちます。ところが，HLが0になったにもかかわらずゼロフラグは立ちません。というより変化しないのです。仮にゼロフラグが立っているとすると，それは以前の演算で立っていたのがそのまま残っているということになります。演算をする際には，表1をよく見て自分が使う命令に対するフラグの変化をよく調べて使うようにしましょう。

誰もが一度はひっかかる命令がINCとDECです。

```
LD     A，FFH
INC    A
```

をやると，当然ゼロフラグが立ちますね。それはよいのですがキャリフラグは変化しません。もっと悲惨なのは「INC　rp」，「DEC　rp」で，こっちはキャリもゼロも変化しないのです。それじゃあどうやって0になったかどうかを判断するのか。これには定石があります。次回の講義の中でお話しすることにしましょう。

今回は算術演算とフラグに重点を置いて進めてきました。フラグがいつセットされ，いつリセットされるのか。よく復習して自分のものにしておいてください。といってもすべて覚える必要はありません。使うときに必ず確認するようにすれば自然と身に付きます。

フラグが実際のプログラムの中でどのような役割を果たしているのか，ということは次講でお話しします。IF文に相当するアセンブリ言語，すなわち「もしキャリが立っていたら～番地に行きなさい」といったたぐいのものを解説するつもりです。次講が理解できればすぐにでも本格的なプログラムが書けるようになります。そのためにも，本講の内容を確実に把握しておいてくださいね。

表1　算術演算とフラグの変化

8ビット演算	キャリ	ゼ　ロ
ADD　A,m	↕	↕
ADC　A,m	↕	↕
SUB　m	↕	↕
SBC　m	↕	↕
INC　r	●	↕
DEC　r	●	↕

16ビット演算	キャリ	ゼ　ロ
ADD　HL,rp	↕	●
ADC　HL,rp	↕	↕
SBC　HL,rp	↕	↕
INC　rp	●	●
DEC　rp	●	●

↕＝変化する　　●＝変化せず

応用プログラム②

あきすとぜねこダンプリスト

先月号の予告どおり，今回は「あきすとぜねこ」のプログラムです。誰もが一度は遊んだことがあるでしょう(でもないかな？)。遊び方を簡単に紹介すると，まず自分の名前を母音だけに直します。名前をローマ字で書いて，a,i,u,e,o を取り出して並べればできます。泉大介なら「いういあいうえ」になりますね。次に，あ→1，い→2，…お→5のように数字に直します。「いういあいうえ」は「2321234」となります。

占いたい相手の名前も数字に直します。小泉今日子なら「5232535」ですね。そうしたら，下のように自分と相手の同じ数字をひとつずつ消していきます。

~~2~~~~3~~~~2~~123~~3~~4　　＝1＋2＋4＝7

5~~2~~~~3~~~~2~~5~~3~~5　　＝5＋5＋5＝15

残った数字を加えて出た数字，7と15がそれぞれの気持ちです。「あきすとぜねこ，あきすと……」と指を折りながら数えていく

と，7は「こ」，15は「あ」です。「こ」は恋人，「あ」は愛している，というわけでこの占いの結果は，「私こと，泉大介は今日子ちゃんのことを恋人だと思っていて，今日子ちゃんは私を愛している」と判断するわけです。

ではプログラムの説明に入ります。

11行目から70行目までは前回のプログラムとほとんど変わりません。違うのは31行と33行でループカウンタをセットするときの値が変わっている程度です。

大きな変更はサブルーチンにあります。まずFILLですが，入力された文字列の最後の文字以降を00_Hで埋める機能のほかに，アスキーコードで入っている数字を数値に直す作業もやっています。たとえば「1」は31_Hとして入っているのですが，これを01_Hに直します。

COUNTは前回，残った文字数を数える役割を負っていましたが，今回は残った数を加えています。

さて，124行にMESTBLというラベルがあるのがわかりますか？

ここが今回の目玉です。先月とはひと味違うテーブル参照の手法をお目にかけましょう。

図2は先月号で使った方法です。EDFMの後ろのダブルクォーテーションの中味は必ず6文字となるようにしましたので，「DEFB 0D_H」と合わせてひとつのメッセージは7バイトとなります。1番目がENGAGEですから，NEGAGEの入っているアドレスに7を足せば次のメッセージの先頭アドレスを指しますね(図中MES+14の所)。それゆえ，Aにメッセージの番号(0～5)を入れ，それを7倍してMESに加えてやることで，表示したいメッセージの入っているアドレスを得ることができたのでした。

今回の方法では，DEFMの後ろのダブルクォーテーションの中味を何文字にしようとまったくかまいません。ではどうやってメッセージのアドレスを得るのか。ここで大きな役割を果たしているのがメッセージテーブルMESTBLです。MESTBLからMES0～MES7のメッセージが始まる先頭アドレスが順に並べて入れてあります。つまり，「LD DE,(MESTBL)」ではDEにMES0のアドレスが入り，「LD DE,(MESTBL+6)」ではDEにMES2のアドレスが入るというわけです。

このようなテーブルを設けることには，次のような意味があります。

1) プログラムを個々のメッセージの長さに関係なく組むことができる。
2) メッセージは整序している必要がなく，またプログラム中の任意の場所に置くことができる。

反面，メッセージの長さに著しい長短の差がなく，かえってテーブルを設けるほうがプログラムが長くなってしまう場合には効力が半減してしまいます。状況と好みに応じて使い分けるのがよいでしょう。

さて，テーブル参照の方法ですが，プログラム中81行からがそのルーチンです。81行に来たときにAにはn(0～7)の値がセットされています。そこでAを2倍してDEにコピーしたあと，MESTBLにDEを足しています(86行)。今HLは「表示すべきメッセージは何番地に入っているよ」という情報の入ったアドレス(MESTBL+2n)を示していますね。そこで，87～89行でその情報をDEにセットします。たとえば81行でA＝1だとすると，89行に来たときにDE＝MES1となるのです。あとはメッセージを表示するだけですね。

一度だけの説明ではわかりづらいかもしれません。そんなときには自分でこういうテーブル参照のルーチンを作ってみてください。そうすれば，きっと意味がわかってもらえると思います。

前回の恋占いに少し手を加えただけで，まったく新しいプログラムになってしまいました。アイデアによってはもっとおもしろいものもできるでしょう。頑張ってみてください。

先月の占いでは岡田有希子ちゃんとの相性はバッチリだった。にもかかわらず，今回の占いではただの「友だち」と出てしまった……。泉大介は失意しております。ま，一喜一憂も恋のうちでしょうね。

というわけで，来月は愛の月，2月。バレンタインデーの2月。愛の相性占い3部作の最後の作品をご披露します。これは，2人の名前を入れると相性は何%と表示してくれる優れものです。それではまた，来月。

図2 テーブル参照(1)

```
MES          DEFM  "NULL  "  DEFB 0DH
MES+7  →     DEFM  "ENGAGE"  DEFB 0DH
MES+14 →     DEFM  "LOVE  "  DEFB 0DH
MES+21 →     DEFM  "BREAK"   DEFB 0DH
                   ⋮
```

図3 テーブル参照(2)

```
                       MES+2  MES+4  MES+6
                         ↓      ↓      ↓
MESTBL  DEFW  MES0 : MES1 : MES2 : MES3
        DEFW  MES4 : MES5 : MES6 : MES7
          ⋮
MES0:   DEFM  " ~ " DEFB 0DH
MES1:   DEFM  " ~ " DEFB 0DH
          ⋮
```

リスト1 あきすとぜねこダンプリスト

```
8000 CD EE 1F 11 12 81 CD E8 :33
8008 1F CD EE 1F 11 45 81 CD :9D
8010 D3 1F 11 23 81 CD E8 1F :7B
8018 CD EE 1F 11 4F 81 CD D3 :5B
8020 1F CD EE 1F 21 45 81 CD :AD
8028 A0 80 21 4F 81 CD A0 80 :FE
8030 21 45 81 06 0A 0E 0A 11 :20
8038 4F 81 1A B7 CA 49 80 BE :F2
8040 C2 49 80 AF 77 12 C3 4E :D4
8048 80 13 0D C2 3A 80 23 05 :44
8050 C2 35 80 21 45 81 CD B7 :E2
8058 80 32 10 81 21 4F 81 CD :01
8060 B7 80 32 11 81 11 35 81 :C2
8068 CD E8 1F 3A 10 81 CD 81 :ED
8070 80 11 3D 81 CD E8 1F 3A :5D
8078 11 81 CD 81 80 CD EE 1F :3A
---------------------------------
SUM: 54 98 5F EF 5E 26 F1 F5 :A4
```

```
8080 C9 B7 CA 8E 80 3D D6 07 :72
8088 D2 86 80 C6 07 3C 87 5F :C7
8090 16 00 21 C1 80 19 5E 23 :12
8098 56 CD E8 1F CD EE 1F C9 :CD
80A0 06 0A 7E B7 CA B0 80 D6 :15
80A8 30 77 23 05 C2 A2 80 C9 :7C
80B0 77 23 05 C2 B0 80 C9 06 :60
80B8 0A AF 86 23 05 C2 BA 80 :63
80C0 C9 D1 80 D6 80 E1 80 E7 :B8
80C8 80 EC 80 F6 80 FD 80 08 :E7
80D0 81 4E 55 4C 4C 0D 41 49 :53
80D8 53 48 49 54 45 49 52 55 :6D
80E0 0D 4B 49 52 41 49 0D 53 :DD
80E8 55 4B 49 0D 54 4F 4D 4F :35
80F0 44 41 43 48 49 0D 5A 45 :05
80F8 4B 4B 4F 55 0D 4E 45 54 :2E
---------------------------------
SUM: CC D2 41 3D 91 3B E9 3F :10
```

```
8100 53 55 52 45 54 53 55 0D :48
8108 4B 4F 49 42 49 54 4F 0D :1E
8110 00 00 49 4E 50 55 54 20 :B0
8118 42 4F 59 27 53 20 4E 41 :13
8120 4D 45 0D 49 4E 50 55 54 :2F
8128 20 47 49 52 4C 27 53 20 :E8
8130 4E 41 4D 45 0D 42 4F 59 :18
8138 20 20 3A 20 0D 47 49 52 :89
8140 4C 20 3A 20 0D 00 00 00 :D3
8148 00 00 00 00 00 00 00 00 :00
8150 00 00 00 00 00 00 00 00 :00
8158 00                      :00
---------------------------------
SUM: 07 00 54 1C 01 1C 86 9A :B4
```

リスト2 あきすとぜねこソースリスト

```
0000                     1 ;   AKISUTO
0000                     2 ;      ZENEKO
0000                     3 ;
0000                     4
8000                     5          ORG     8000H
8000                     6          ;
8000                     7 GETL:    EQU     1FD3H
8000                     8 MSG:     EQU     1FE8H
8000                     9 LETNL:   EQU     1FEEH
8000                    10
8000                    11 START:
8000 CD EE 1F           12          CALL    LETNL
8003 11 12 81           13          LD      DE,MESB
8006 CD E8 1F           14          CALL    MSG
8009 CD EE 1F           15          CALL    LETNL
800C 11 45 81           16          LD      DE,BN
800F CD D3 1F           17          CALL    GETL      ; INPUT BOY'S NAME
8012 11 23 81           18          LD      DE,MESG
8015 CD E8 1F           19          CALL    MSG
8018 CD EE 1F           20          CALL    LETNL
801B 11 4F 81           21          LD      DE,GN
801E CD D3 1F           22          CALL    GETL      ;INPUT GIRL'S NAME
8021 CD EE 1F           23          CALL    LETNL
8024                    24          ;
8024 21 45 81           25          LD      HL,BN
8027 CD A0 80           26          CALL    FILL
802A 21 4F 81           27          LD      HL,GN
802D CD A0 80           28          CALL    FILL
8030                    29          ;
8030 21 45 81           30          LD      HL,BN
8033 06 0A              31          LD      B,10
8035                    32 CMPR1:
8035 0E 0A              33          LD      C,10
8037 11 4F 81           34          LD      DE,GN
803A                    35 CMPR2:
803A 1A                 36          LD      A,(DE)
803B B7                 37          OR      A
803C CA 49 80           38          JP      Z,NOTEQ
803F BE                 39          CP      (HL)
8040 C2 49 80           40          JP      NZ,NOTEQ
8043 AF                 41          XOR     A
8044 77                 42          LD      (HL),A
8045 12                 43          LD      (DE),A
8046 C3 4E 80           44          JP      SAME
8049                    45 NOTEQ:
8049 13                 46          INC     DE
804A 0D                 47          DEC     C
804B C2 3A 80           48          JP      NZ,CMPR2
804E                    49 SAME:
804E 23                 50          INC     HL
804F 05                 51          DEC     B
8050 C2 35 80           52          JP      NZ,CMPR1
8053                    53          ;
8053 21 45 81           54          LD      HL,BN
8056 CD B7 80           55          CALL    COUNT
8059 32 10 81           56          LD      (BOY),A
805C 21 4F 81           57          LD      HL,GN
805F CD B7 80           58          CALL    COUNT
8062 32 11 81           59          LD      (GIRL),A
8065                    60          ;
8065 11 35 81           61          LD      DE,MESB1
8068 CD E8 1F           62          CALL    MSG
806B 3A 10 81           63          LD      A,(BOY)
806E CD 81 80           64          CALL    PRTSUB
8071 11 3D 81           65          LD      DE,MESG1
8074 CD E8 1F           66          CALL    MSG
8077 3A 11 81           67          LD      A,(GIRL)
807A CD 81 80           68          CALL    PRTSUB
807D CD EE 1F           69          CALL    LETNL
8080 C9                 70          RET
8081                    71
8081                    72 PRTSUB:
8081 B7                 73          OR      A
8082 CA 8E 80           74          JP      Z,PRT
8085 3D                 75          DEC     A
8086                    76 PRTSB1:
8086 D6 07              77          SUB     7
8088 D2 86 80           78          JP      NC,PRTSB1
808B C6 07              79          ADD     A,7
808D 3C                 80          INC     A           ; A / 7 + 1
808E                    81 PRT:
808E 87                 82          ADD     A,A         ; A *= 2
808F 5F                 83          LD      E,A
8090 16 00              84          LD      D,0
8092 21 C1 80           85          LD      HL,MESTBL   ; MESSAGE TABLE
8095 19                 86          ADD     HL,DE
8096 5E                 87          LD      E,(HL)
8097 23                 88          INC     HL
8098 56                 89          LD      D,(HL)
8099 CD E8 1F           90          CALL    MSG
809C CD EE 1F           91          CALL    LETNL
809F C9                 92          RET
80A0                    93
80A0                    94 FILL:
80A0 06 0A              95          LD      B,10
80A2                    96 FILL1:
80A2 7E                 97          LD      A,(HL)
80A3 B7                 98          OR      A
80A4 CA B0 80           99          JP      Z,FILL2
80A7 D6 30             100          SUB     '0'
80A9 77                101          LD      (HL),A
80AA 23                102          INC     HL
80AB 05                103          DEC     B
80AC C2 A2 80          104          JP      NZ,FILL1
80AF C9                105          RET
80B0                   106          ;
80B0                   107 FILL2:
80B0 77                108          LD      (HL),A
80B1 23                109          INC     HL
80B2 05                110          DEC     B
80B3 C2 B0 80          111          JP      NZ,FILL2
80B6 C9                112          RET
80B7                   113
80B7                   114 COUNT:
80B7 06 0A             115          LD      B,10
80B9 AF                116          XOR     A
80BA                   117 COUNT1:
80BA 86                118          ADD     A,(HL)
80BB 23                119          INC     HL
80BC 05                120          DEC     B
80BD C2 BA 80          121          JP      NZ,COUNT1
80C0 C9                122          RET
80C1                   123
80C1 D1 80 D6 80 E1 80 E7  124 MESTBL: DEFW  MES0:MES1:MES2:MES3
80C8 80
80C9 EC 80 F6 80 FD 80 08  125         DEFW  MES4:MES5:MES6:MES7
80D0 81
80D1                   126
80D1 4E 55 4C 4C 0D    127 MES0:   DEFM    "NULL"         DEFB 0DH
80D6 41 49 53 48 49 54 45  128 MES1:   DEFM    "AISHITEIRU"   DEFB 0DH
80DD 49 52 55 0D
80E1 4B 49 52 41 49 0D 129 MES2:   DEFM    "KIRAI"        DEFB 0DH
80E7 53 55 4B 49 0D    130 MES3:   DEFM    "SUKI"         DEFB 0DH
80EC 54 4F 4D 4F 44 41 43  131 MES4:   DEFM    "TOMODACHI"    DEFB 0DH
80F3 48 49 0D
80F6 5A 45 4B 4B 4F 55 0D  132 MES5:   DEFM    "ZEKKOU"       DEFB 0DH
80FD 4E 45 54 53 55 52 45  133 MES6:   DEFM    "NETSURETSU"   DEFB 0DH
8104 54 53 55 0D
8108 4B 4F 49 42 49 54 4F  134 MES7:   DEFM    "KOIBITO"      DEFB 0DH
810F 0D
8110                   135
8110 00                136 BOY:    DEFB    0
8111 00                137 GIRL:   DEFB    0
8112                   138
8112 49 4E 50 55 54 20 42  139 MESB:   DEFM    "INPUT BOY'S NAME"
8119 4F 59 27 53 20 4E 41
8120 4D 45
8122 0D                140         DEFB    0DH
8123 49 4E 50 55 54 20 47  141 MESG:   DEFM    "INPUT GIRL'S NAME"
812A 49 52 4C 27 53 20 4E
8131 41 4D 45
8134 0D                142         DEFB    0DH
8135 42 4F 59 20 20 3A 20  143 MESB1:  DEFM    "BOY  : "
813C 0D                144         DEFB    0DH
813D 47 49 52 4C 20 3A 20  145 MESG1:  DEFM    "GIRL : "
8144 0D                146         DEFB    0DH
8145                   147
8145 00 00 00 00 00 00 00  148 BN:     DEFS    10
814C 00 00 00
814F 00 00 00 00 00 00 00  149 GN:     DEFS    10
8156 00 00 00
```

ますますツメターイBASIC塾 第2講

グラフィクス&グラフ

Takahara Hideki
高原ひでき

85年10月号に掲載したBASIC塾第1期第6講では「音と光のファンタジー」と称してパソコングラフィクス入門をしましたが，今回はもうちょっと突っ込んでみたいと思います。パターン絵画の描き方とグラフの作成法を研究しましょう。今月の内容はやや難しいので，中級者の方も臨時受講生として参加してください。スペシャルプログラムとして汎用グラフ作成ソフト「ぐらふくん」を作ってみました。

1 基礎知識編

'85年10月号の復習をとりあえずすませましょう。パソコングラフィクスは，キャラクタ画面の操作と違って覚えることが多いので，改めてざっと一読してください。なお，本文中でMZ-1500用S-BASICは(S15)と表すことにします。

① グラフィクス画面の用意

BASIC起動時には絵は描けませんので，次の命令文がプログラムのはじめに必ず必要です。

(Hu) SCREEN 0, 0, 0 : WIDTH 80
(S15) INIT "CRT : G"

② 点を打つ

パソコンの画面は横320×縦200ドットの小さな網目になっています(MZ-1500以外は640×200もある)。この座標，すなわちポジションは（右に何個目，下に何個目）で表します。左上隅が原点で(0, 0)，右下が(319, 199)です。点の打ち方は，

(Hu) PSET (X, Y, C)
(S15) SET [C] X, Y

Cは色番号で0の透明(黒)，1の青から7の白まで8色あります。

③ 2点間に直線を引く

点(X1, Y1)と(X2, Y2)の間を直線で結ぶには，

(Hu) LINE(X1, Y1)－(X2, Y2), PSET, C
(S15) LINE[C]X1, Y1, X2, Y2

とします。

④ 長方形(水平に限る)

対角線座標が(X1, Y1)と(X2, Y2)である長方形は，

(Hu) LINE(X1, Y1)－(X2, Y2), PSET, C, B
(S15) BOX[C]X1, Y1, X2, Y2

です。これが，

(Hu) LINE(X1, Y1)－(X2, Y2), PSET, C, BF
(S15) BOX[C]X1, Y1, X2, Y2, 色番号

になると(色番号)で四角形の内部をぬりつぶします。

⑤ 円

(Hu) CIRCLE(X, Y), R, C
(S15) CIRCLE[C]X, Y, R

これは点(X, Y)が中心で，半径がRドットの円の描き方です。

以上で基礎段階が終わったことにします(さすがにツメタイ！)。なお，今回はあまりBASIC間の移植方法を指摘しませんので，この項を見ながら変換しましょう。この他は違いはでません。

2 COSとSIN

この項は数学，とくにSIN, COS(サイン，コサイン)がわからない人にとっては修羅場となりますが，意外とこれを読むことで，数学ができるようになるかもしれませんので心してください。

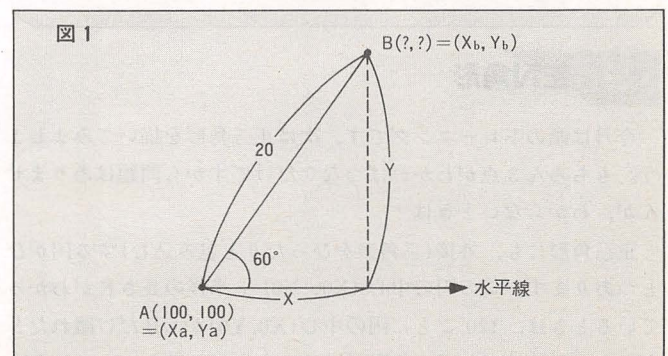
まず，図1を見てください。点Bは点Aから20ドット離れています。このときの点Bの座標はどうなるでしょうか。

横座標は $X_b = X_a + X = 100 + X$
縦座標は $Y_b = Y_a - Y = 100 - Y$ (下向きがプラス)

これはわかりますね。それではXとYはどうやれば求まるのでしょうか。こんなときに使うのがSIN,COSなのです。

いま直線ABと水平線の間の角度が60°です。このときに，直線ABを水平方向に縮めるためにかける数がCOS(角度)，上下方向に縮めるためにかける数がSIN(角度)なのです。

ですから,

X＝AB＊COS (60°)＝20＊COS (60°)

Y＝AB＊SIN (60°)＝20＊SIN (60°)

なのです。ところがここでさらに，角度の変換をしなければなりません。

パソコンではほとんどの場合,「ラジアン」という角度の単位を使っています。これは180°が円周率π＝3.1415…に相当する不思議な単位系なのです。なんでこうなのかは私も詳しくは知りませんが要は180°＝πラジアンなのです。そこで

(Hu)　ラジアン＝(角度)＊PAI(1)/180

(S15)　ラジアン＝(角度)＊π/180

という式で変換しなければならない，とだけ覚えてください。

※注：HuBASICでもπが使えます。好きな方を使ってください。

結局，図1の点Bの座標はHuBASICでは

X_b＝100＋20＊COS (60＊PAI(1)/180)＝110

Y_b＝100−20＊SIN (60＊PAI(1)/180)＝82.679492

となるのです。

まとめ

点（X_a, Y_a）から右方向に対してQ度の角度を作りRドット離れた点Bの座標（X_b, Y_b）は

X_b＝X_a＋R＊COS(Q＊PAI(1)/180)

Y_b＝Y_a－R＊SIN (Q＊PAI(1)/180)

※注：数学ではY座標の方向が逆なのでSINの前の符号は−ではなく＋になります。くれぐれも注意してください。

PRO1．円を描く

CIRCLE命令を使わずに円を描いてみよう。

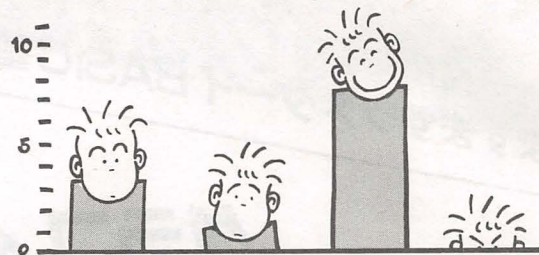
もういちど図1をみてみよう。角度は60°になっているが，これが0°から360°までぐるっと1周すると点Bの跡はどうなるか。そう，円になるのです。図1の値を使ってプログラムを使ってみもう一度図1をみてみましょう。先ほどのまとめとまったく同じでしょ。SIN,COSなんて簡単なんです。

<PRO1.>円を描く(HuBASIC)

```
100 XA=100:YA=100:R=20
110 SCREEN 0,0,0
120 FOR Q=0 TO 360
130 XB=XA+R*COS(Q*PAI(1)/180)
140 YB=YA-R*SIN(Q*PAI(1)/180)
150 PSET(XB,YB,1)
160 NEXT Q
```

3　正N角形

今月は頭のトレーニングです。次は正三角形を描いてみましょう。もちろん3点がわかればつなぐだけですから問題はありませんが，わからないときは？

正三角形にも，外接(三角形をぴったりと包み込む)する円がひとつあります。この円の中心（X0，Y0）と半径の長さRがわかっているときは，120°ごとに円の中心(X0,Y0)からRだけ離れた3点をそれぞれ結ぶと正三角形が描けます。この場合は中心から右に引いた軸と半径との中心角はA°，A°＋120°，A°＋240°になりますから，3点の座標を計算して，順に直線でつないでいくわけで，このプログラムがPRO2です。

<PRO2.>(HuBASIC)

```
100 CLS0
110 SCREEN 0,0,0
120 DEF FNK(Z)=Z*PAI(1)/180
130 X0=100:Y0=100:R=50
140 N=3:A=90:C=1
150  X=X0+R*COS(FNK(A))
160  Y=Y0-R*SIN(FNK(A))
170 FOR Z=1 TO N
180  X1=X0+R*COS(FNK(A+Z*360/N))
190  Y1=X0-R*SIN(FNK(A+Z*360/N))
200  LINE(X,Y)-(X1,Y1),PSET,C
210 X=X1:Y=Y1
220 NEXT Z
```

正N角形

PRO3．外接円の中心が(X0，Y0)，半径Rのときの正N角形を描くプログラムを考えよう。

答はPRO2です。Aは初期角，Cはカラー番号です。

何ですって，ミスプリントではありませんよ。先ほどと同じくPRO2が答です。つまり140行のNの値を入れ替えるだけでいいように，始めから作ってあるのだ。ハッハッハ。

このように値を変数に入れておくと便利なのです。

ところで中心の座標がわからなくても，点が進む方向がわかれば多角形は描けます。すなわち，点が図形をなぞっていくときは，その点は何かしらのベクトルを持っています。ベクトルとは横方向と縦方向に進む長さを表す量のことですが，このベクトルを使って正N角形を描く方法をPRO3として掲載しましょう。

<PRO3.>正N角形(HuBASIC)

```
100 CLS:SCREEN 0,0,0
110 PRINT "* ｶｲｼﾃﾝｦ ｷﾒﾙ."
120 INPUT " X=";PX
130 INPUT " Y=";PY
140 INPUT "* ﾅﾝｶｯｹｲ ﾃﾞｽｶ:";N
150 INPUT "* ﾍﾝﾉ ﾅｶﾞｻ=";R
160 INPUT "* COLOR=";C
170 CLS
180 A=2*PAI(1)/N
190 FOR Z=1 TO N
200  VX=COS(A*Z):VY=SIN(A*Z)
210  FOR X=0 TO R
220   PX=PX+VX:PY=PY-VY
230   PSET(PX,PY,C)
240  NEXT X
250 NEXT Z
260 GOTO 110
```

この場合は開始点の座標さえわかればいいのです。あとは200行で計算したベクトルに応じて1辺ずつ順に点が跡を残しながら正N角形を描きます。そうです！ LOGOのタートルグラフィクスと同じですよ。

PRO4. スピログラフ

板にあいた大きな穴に小さな歯車をあてて，その歯車にあいている小さな穴にボールペンを差し込んでボールペンを動かすと，不思議なことに，美しいグラフィクスが描けますね。

無謀にもこの不思議な図形をパソコングラフィクスで描きます。それがPRO4です。MZ-1500ユーザーへの特別サービスとして，いつもと逆にしてみました。

〈PRO4.〉スピログラフ(MZ-1500)

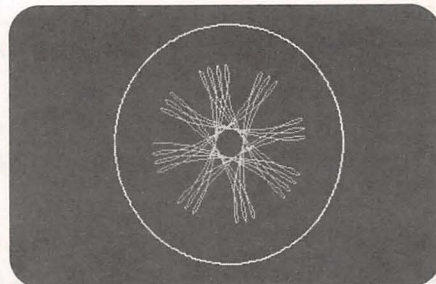
```
100 REM ***** ｽﾋﾟﾛ ｸﾞﾗﾌ *****
110 X0=160:Y0=100:R0=99:REM ｿﾄﾉ ｴﾝ
120 R1=60:REM ﾅｶﾉ ｴﾝ ﾉ ﾊﾝｹｲ
130 XX=50:REM ﾅｶﾉ ｴﾝﾆ ｳﾂ ﾃﾝ(<R1)
140 C=2:REM   COLOR
150 N=10:REM  ｶｲﾃﾝｽｳ
160 CLS 3:INIT "CRT:G"
170 CIRCLE [7]X0,Y0,R0
180 CIRCLE [7]X0+R0-R1,Y0,R1-1
190 LINE[7]X0+R0-R1,Y0,X0+R0,Y0
199  RT=19:LT=20:REM CHR-CODE
200  CCOLOR,,3:CURSOR 0,23
210  PRINT "* ｶｲｼﾃﾝﾉ ｲﾁｦ[<->]ﾃﾞｷﾒﾙ.OK=[CR]"
220  SET[0]X0+R0-R1+XX,Y0
230  GET Z$:IF Z$="" THEN 230
240  IF Z$=CHR$(13) THEN 340
250  IF Z$=CHR$(RT) THEN DX=1:GOTO 280
260  IF Z$=CHR$(LT) THEN DX=-1:GOTO 280
270  GOTO 230
280  SET[7]X0+R0-R1+XX,Y0
290  XX=XX+DX
300  IF XX<1 THEN XX=1
310  IF XX>R1 THEN XX=R1
320  GOTO 220
330 REM === ROTATION START ===
340 CLS1
350 CIRCLE [0]X0+R0-R1,Y0,R1-1
360 LINE[0]X0+R0-R1,Y0,X0+R0-1,Y0
370 QX=X0+R0-R1+XX:QY=Y0
380 FOR Z=0 TO 360*N STEP 6
390 A=Z*π/180
400 PX=X0+(R0-R1)*COS(A)
410 PX=PX+XX*COS(-A*R0/R1)
420 PY=Y0-(R0-R1)*SIN(A)
430 PY=PY-XX*SIN(-A*R0/R1)
440 LINE[C]PX,PY,QX,QY
450 QX=PX:QY=PY
460 NEXT Z
470 REM CIRCLE [0]X0,Y0,R0
```

★HuBASIC変更点(一部)

```
199  RT=28:LT=29
200  COLOR 3:CURSOR 0, 23
230  Z$=INKEY$:IF Z$=""THEN 230
390  π→PAI(1)
```

110行から150行で必要な値を設定しています。(X0, Y0)が板の円の中心座標。R0が半径。小さな歯車の半径がR1です。XXは歯車にあける穴の中心からの距離，そしてCが色番号です。あらかじめ初期設定がしてありますので自由に値を変えて楽しんでください。

199行から320行まで字下げがしてありますが，この字下げした行はすべて，めんどくさい人は省略してけっこうです。なぜか？それは実際に作ってみればわかります。

ごく簡単に仕組みを説明しますと，歯車を円周に沿ってA°だけ反時計回りに回すと，歯車は逆の時計回りにA°の(R0/R1)倍，すなわち半径比倍回ることになります。この原理で380行から460行の間でボールペンの座標の動きを計算してプロットしています。

スピログラフ

4 グラフを描く

いよいよ今月のメインテーマであるグラフ作成に突入します。企業でパソコンを事務処理に使っているときは，グラフ作成は欠かせないのですが，それに比べて家で個人がパソコンを使うときはグラフは縁遠い存在です。あなた，一度でもパソコンでグラフを描いたことがありますか。そこで私がコーチするのです。

棒グラフを描く予備知識

図2を描く方法を考えます。

(いまわかっていること) 原点Oの座標，OPの長さH，OQの長さ3＊W，阿部君の身長Acm，加藤さんの身長Bcm(<A)。

1. 座標軸を描く

まずOPを引きましょう。Oはいいとして，Pの座標を求めます。OPの距離はHとわかっていますので，縦座標はY0－Hとなります。横座標は同じです。結局，点P＝(X0, Y0－H)です。同様に点Q＝(X0＋3＊W, Y0)となります。これをHuBASICで描くと

LINE(X0,Y0)－(X0,Y0－H),PSET,7 (直線OP)

LINE(X0,Y0)－(X0＋3＊W,Y0),PSET,7 (直線OQ)

となります。

2. 阿部君の身長を描く

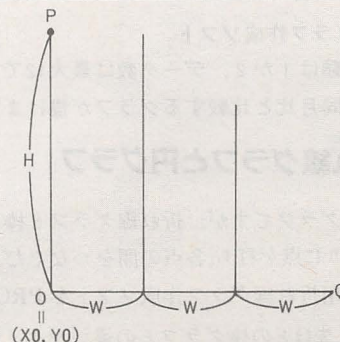
次に阿部君の身長を描き込むわけですが，このグラフ上の長さをOPと同じにします。これも簡単で，A1＝(X0＋W,Y0)，A2＝(X0＋W,Y0－h)となりますからHuBASICでは

LINE(X0＋W,Y0)－(X0＋W,Y0－h),PSET,7

となります。

3. 加藤さんの身長を描く

これが一番やっかいなのです。点B1,B2とも縦座標はX0＋2＊W

図2

だとわかりますが，高さがわかりません。

ここで阿部君，加藤さんとも本当の身長がA, Bだとわかっていますから比の値をとると

(阿部君の身長)：(加藤さんの身長)＝A：B

＝(阿部君のグラフでの長さH)：(加藤さんのグラフでの長さZ)

となりますのでA：B＝H：Zです。わからないのはZだけですが，比例式では(内項の積)=(外項の積)すなわちA＊Z＝B＊Hですので，結局Z＝H＊A/Bとなります。したがってBASICの文は，

LINE(X0＋2＊W, Y0)－(X0＋2＊W, Y0－H＊A/B),

PSET, 7

となります。

これで図2の絵が描けましたね。

4. 棒グラフにするには

実際には棒グラフという通り，棒で量を表現しますので，これまで直線だった阿部君の身長と加藤さんの身長を，今度は棒に変えなくてはなりません。しかし，これは簡単です。棒の太さVがわかればその太さ分だけ点A1を右にずらせてLINE, BF文を使えばいいのですからこの場合，阿部君の身長を青い棒で表現すると

LINE(X0＋W,Y 0)－(X0＋W＋V,Y0－H),PSET,1,BF

となります。加藤さんの場合も同様にVを第2項に加えるだけでいいのです。

5. 実際の応用

ここでは，阿部君の身長が加藤さんの身長よりも高いことがわかりましたが，3人目以降が登場してくると，最も高い人をパソコンで選出しなければなりません。これは次のようにします。

```
DM=D(1)
FOR  X=2  TO  N
IF  DM<D(X)  THEN  DM=D(X)
NEXT  X
```

Nは人数，D(X)は各自の身長です。そして先ほどと同様に

Z=H＊D(X)/DM

としてグラフ上の長さを調べます。グラフ作りもFOR～NEXT文で行います。

さらに，比較棒グラフでは2種類以上のデータがでてきますので，各データはD（データの種類，データの番号）のような配列に納めます。では今の説明がサッパリわからなかった数学オンチの人でも，楽に使える汎用棒グラフ作成ソフトをプレゼントしましょう。

PRO 5．棒グラフ作成ソフト

データの種類は1か2，データ数は最大12です。たとえば毎日の貯金を前年同月比と比較するグラフが描けます。

5 折れ線グラフと円グラフ

次は折れ線グラフですが，折れ線グラフと棒グラフの違いは棒をたてる代わりに点を打ち，各点の間をつなぐだけです。

これまた汎用折れ線グラフ作成ソフトをPRO6として掲載しておきますので，先ほどの棒グラフとの違いをよく観察してみてください。

〈PRO5.〉 棒グラフ(HuBASIC)

```
100 REM ***** DATA-INPUT *******
110 INPUT "* ﾃﾞｰﾀ ﾉ ｼｭﾙｲﾊ (1/2) :";Y
120 INPUT "* ﾃﾞｰﾀ ﾉ ｶｽﾞﾊ (1-12):";T
130 DIM D(Y,T):CLS
140 FOR Z=1 TO Y
150 CURSOR (Z-1)*20,0:PRINT "* ITEM No.";Z
160 FOR X=1 TO T:GOSUB 210:NEXT X
170 CURSOR 0,22:INPUT "* MISS No.(ﾅｼ=0):";X
180 IF X=0 THEN 200
190 IF X>12 THEN 160
195 GOSUB 210:GOTO 170
200 NEXT Z:GOTO 249
210 CURSOR (Z-1)*20,X:PRINT "* DATA";X;
220 INPUT "=";D(Z,X)
230 RETURN
249 REM
400 REM ***** BAR-GRAPH *****
410 SCREEN 0,0,0:CLS
420 WO=240:W=WO/T:H=128
430 XO=40:YO=160:DM=0:C=3
440 LINE(XO,YO)-(XO,YO-H),PSET,1
450 LINE(XO,YO)-(XO+WO,YO),PSET,1
460 FOR Z=1 TO Y
470  FOR X=1 TO T
480   IF D(Z,X)>DM THEN DM=D(Z,X)
490 NEXT X,Z
500 DN=INT(LOG(DM)/LOG(10))
501 DL=INT(DM/10^DN):DK=DL*10^DN
502 FOR X=0 TO DK STEP 10^DN
503  DJ=YO-H*X/DM
504  LINE(XO-5,DJ)-(XO,DJ),PSET,7
505 NEXT X
510 FOR Z=Y TO 1 STEP -1
520 GOSUB 550
530 NEXT Z
540 GOTO 999
550 FOR X=1 TO T:REM --- SUB BAR -----
560 LINE(XO+(X-.5)*W+Z*3,YO)-(XO+X*W+Z*3,YO
    -D(Z,X)*H/DM),PSET,C,BF
570 NEXT X:C=1
580 RETURN
999 REM END
```

(注)次の折れ線グラフPRO6.と円グラフPRO7.について，100行から249行まではこのPRO5.をそのまま使います。

円グラフはちょっと様子が違います。次の要領で描きます。

1. 円を描く

まず円の中心の座標O（X0，Y0）と半径Rを決めます。そしてCIRCLE文で円を描きます(870行)。

2. データの合計を求める

棒グラフや折れ線グラフは各データの量を比較するグラフですが，円グラフは各データが全体のうちどれだけを占めるかを調べるものです。ですから，各データの合計をFOR～NEXT文で求めなくてはなりません。

この合計値Dが円の全周2＊PAI(1) ラジアン値に相当するので，各データに割り当てられるラジアン値は

D(X)/D＊2＊PAI(0)

となります。

3. 円に線を引く

そして円の中心から各値に割り当てられた角度まで，順に線を引いていきます。ところで基線は中心から真上に引いた線ですから，角度では90°＝π／2ラジアンとなるので，この値を先ほどの割り当て値に加えて，あとはLINE文で半径を引いていきます。

この手法についてはPRO6：汎用円グラフパッケージを調べてください。

〈PRO6.〉折れ線グラフ(PRO5.の100～249行を追加)

```
600 REM ***** LINE-GRAPH *****
610 SCREEN 0,0,0:CLS
620 WO=240:W=WO/T:H=128:C=1
630 XO=40:YO=160:DM=0
640 LINE(XO,YO)-(XO,YO-H),PSET,7
650 LINE(XO,YO)-(XO+WO,YO),PSET,7
660 FOR Z=1 TO Y
670 FOR X=1 TO T
680 IF D(Z,X)>DM THEN DM=D(Z,X)
690 NEXT X,Z
700 DN=INT(LOG(DM)/LOG(10))
702 DL=INT(DM/10^DN):DK=DL*10^DN
704 FOR X=0 TO DK STEP 10^DN
706 DJ=YO-H*X/DM
708 LINE(XO-5,DJ)-(XO,DJ),PSET,7
709 NEXT X
710 FOR Z=1 TO Y:GOSUB 730:NEXT Z
720 GOTO 999
730 FOR X=1 TO T:REM --- SUB LINE ---
740 PX=XO+(X-.5)*W:PY=YO-D(Z,X)*H/DM
750 IF X=1 THEN QX=PX:QY=PY
760 CIRCLE(PX,PY),1,C
770 LINE(PX,PY)-(QX,QY),PSET,C
780 QX=PX:QY=PY
790 NEXT X:C=3
799 RETURN
999 REM END
```

〈PRO7.〉円グラフ(PRO5.の100～249行を追加)

```
800 REM ***** CIRCLE-GRAPH *****
810 SCREEN 0,0,0:CLS 4
820 XO=150:YO=100
830 R=50:D=0
840 FOR X=1 TO T
850 D=D+D(1,X)
860 NEXT X
870 CIRCLE(XO,YO),R,7
880 FOR Z=0 TO T-1
890 DO=DO+D(1,Z):A=PAI(1)*DO*2/D
900 X=XO+R*SIN(A):Y=YO-R*COS(A)
910 LINE(XO,YO)-(X,Y),PSET,7
920 NEXT Z
930 GOTO 999
999 REM END
```

6 今月のプログラム：ぐらふくん

ここまで棒グラフ，折れ線グラフ，円グラフを描く汎用ソフトを紹介してきましたが，集大成となるのがこの「ぐらふくん」です。データを入力後，メニューとして表示されるグラフスタイルを選ぶと，そのグラフを描いて終了します。せっかく入力したデータですから，記録しておきたいときは，ファイルとしてテープやディスク，QDに記録できます。

ところで初の試み！　なんと，この「ぐらふくん」は先ほどから研究してきたPRO 5～7と残りPRO7の合成体なのです。もしPRO4からPRO6までをセーブしていた人は次の方法で合成してください。

> **MERGE**
> テープやディスクにセーブしたプログラムF＄を，いま使っているプログラムに合成するときの命令。
> MERGE “ファイル名”

(方法)マージするプログラムをSAVE “File1”，Aとしてセーブする。次にマージされるほうのプログラムをロードしてMERGE “File1”で２つのプログラムを合体させる。なお，この機能のないBASICの場合は必殺画面マージ（画面を使ってマージすること）を使おう。

いやはや，今月は大盛りというか２回分というか，あまりの盛りだくさんの内容で私は疲れてしまいました。

今月のプログラム：ぐらふくん
このプログラムにPRO5.～7.をMERGEしてください。

```
105 INPUT "> ﾌｧｲﾙｶﾗ ﾖﾐﾏｽｶ(Y/N):";ANS$
106 IF ANS$="Y" THEN GOSUB 1200:GOTO 250
250 REM ***** MENU *****
260 CLS 4:CURSOR 5,0:PRINT "* GRAPH-STYLE
    MENU *"
270 CURSOR 0,2:PRINT "1. BAR-GRAPH"
280 CURSOR 0,4:PRINT "2. LINE-GRAPH"
290 CURSOR 0,6:PRINT "3. CIRCLE-GRAPH"
300 CURSOR 0,8:PRINT "4. BAR & LINE-GRAPH"
310 CURSOR 0,12
320 INPUT "* SELLECT No.=";ANS
330 IF (ANS<1)OR(ANS>4) THEN 310
340 ON ANS GOTO 400,600,800,400
390 GOTO 310
509 IF ANS=4 THEN 950
950 REM *** BAR & LINE *****
960 Z=1:GOSUB 550
970 Z=2:GOSUB 730
1000 REM ***** OPTION *****
1010 INPUT "> ﾌｧｲﾙ ｼﾏｽｶ(Y/N):";ANS$
1020 IF ANS$<>"Y" THEN 1120
1030 INPUT "* FILE NAME =";F$
1040 OPEN "O",#1,F$
1050 PRINT#1,Y
1060 PRINT#1,T
1070 FOR U=1 TO Y
1080 FOR V=1 TO T
1090 PRINT#1,D(U,V)
1100 NEXT V,U
1110 CLOSE#1
1120 END
1200 INPUT "* FILE NAME =";F$
1210 OPEN "I",#1,F$
1220 INPUT#1,Y
1230 INPUT#1,T
1240 DIM D(Y,T)
1250 FOR U=1 TO Y
1260 FOR V=1 TO T
1270 INPUT#1,D(U,V)
1280 NEXT V,U
1290 CLOSE#1:RETURN
```

★S-BASIC 1500用変更点(一部)

```
1040 WOPEN #1, F$
1210 ROPEN #1, F$
```

全機種共通

詩人になりたいひとのプログラム.

ザ・コピーライター

Kudo Makoto
工藤　誠

近ごろアイデアが貧困でとお悩みのあなた，発想の転換に最適のプログラムがこれです。言葉と言葉の組み合わせは無限に近く，パソコンによって次々に生み出されるコピーの連続に，今までの固定観念はぶっとびとなるでしょう。

星 は 淋しく 冷たい
淋しい 雨 よ
輝き は 若い 初恋 のようだ
淋しい 雨 よ
空 は 哀しく 優しい
過去 は 銀河 を 戯れる
蛍 のように セーラー服 は 遊ぶ
花 は 淋しい 涙 のように
月 が 蒼く 冷たい
過去 は 銀河 を 引き裂く
星 は 蒼く 眠る
夜空 の 虚しい 輝き よ

X1 NEW BASIC作

これはいわゆるコピーや詩を作成するプログラムで，ようするに品詞のパターンを作ってその品詞ごとにランダムに言葉を選ぶプログラムです。使い方は簡単で，メニューの選択は詩の作成と終了のみ。詩を画面に表示したあとは，プリンタ出力の有無を聞いてきますので，好きな行，あるいはすべての行を印字することができます。

このプログラムでは，作られた文中の単語間には，何も関係はありません。すべてランダムに選ばれます。当初は，それぞれの単語に指向性をつけることも考えました。たとえば，「雨」に対して，「降る」などの動詞や「強い」などの形容詞が対応するような形です。しかし，そのように使う単語に制限をつけると，同じような文句ばかり並んだりしますし，何よりも人間の常識を超えるような文句は出てこなくなります。そこで単語はすべてランダムに選ぶようにしました。半分ぐらいが，わけのわからない文になります。「瞳はまったりと眠る」「レモンの逢いたい波」などは，なんだかよくわかりません。しかし，ランダムだからこそはっとするような文句（コピー）も出たりするものです。

具体的にどのように使えるかを考えてみましょう。まず，単語データに，およそ詩には似つかわしくない単語を多くまぜて，出てくるとんちんかんな文句を数人で見て笑いころげる。これは健康にいいと思います。ある特定の単語をデータに多く入れておくと，出現回数が多くなりますので，ある一定のテーマについての文句を作ることができます。これを利用して，標語などを作ることも可能です。たとえば，車，走る，歩く，危い，などの単語を多くすると，交通安全の標語が作れそうですね。また，ある商品の特長をデータに多く入れておくと，意外な名コピーが作れるかもしれません。近ごろのコピーはわけのわからないのが多いので，十分このプログラムでも対抗できます。というのはもちろん冗談です。しかし，好きな行を1行だけ印字できるというのは，はっきり言ってコピー作成を意識してつけた機能です。ようするに，このプログラムは10000行以下のデータを変えることによって，作る文の雰囲気を変えることができるのです。イメージが暗い単語を多くしますと，暗い詩ができますし，逆だと明るい詩になります。また，有名な詩人の詩の中からデータを選んで入力すれば，たとえば北原白秋風とか，高村光太郎風の詩ができるかもしれません。また，何かのアイデアにつまったときにも使えます。正常な文中ではまず絶対に組むことがない単語がひとつの文中に続いて出たりします。たとえば，「花は飛ぶ」などの類です。これなどは発想の転換をはかったりするのに使えそうです。

次に，データの拡張について述べます。データは10000行以下に，名詞，動詞，形容詞，形容動詞，副詞，その他の順に入っています。各品詞のデータの終わりには，“＊”を入れておいてください。形容詞は，「〜い」の形で，形容動詞は「〜だ」の形で入れておいてください。この2つは，以下に続く言葉によって語尾を変化させます。動詞は変化させていないので「生きる」「生きている」のように，いろいろな形で入れておいてください。

サンプル

```
ヨゾ゛ラ ハ セーラーフク ダ゛
ネムル カコ ハ カコ ダ゛
ツメタイ セーラーフク
カガ゛ヤキ ガ゛ ソラ ヲ ユックリ オモウ
モエル ハツコイ ヨ
トキメク カガ゛ヤキ ヨ
ソラ ハ コウコウト アイタイ
ヨゾ゛ラ ノヨウニ ショウジ゛ョ ハ ナミダ゛ ダ゛
セーラーフク ハ ワクワク クライ
オモイデ゛ ハ ワガ゛ママダ゛
ギ゛ンガ゛ ガ゛ ナミ ヲ トキメク
コイシイ ヨゾ゛ラ
ヨゾ゛ラ ノヨウニ ツキ ハ ショウネン ダ゛
ハツコイ ガ゛ マッタリト コイシイ
ナガ゛レル ユウヒ ハ トブ゛
ハツコイ ノヨウニ ハツコイ ハ タカイ
ツキ ハ ハナ ヲ テラシテイル
ワラウ カガ゛ヤキ ハ メザ゛メル
ナミダ゛ ガ゛ ナヤマシイ ヒトミ
ツキ ハ スギ゛サッタ
ホタル ハ フシギ゛ナ ショウネン ノヨウダ゛
ショウジ゛ョ ハ シロイ
ナミダ゛ ハ アカルイ
ナミ ハ カナシイ
ギ゛ンガ゛ ハ チッポ゜ケニ アソブ゛
ソラ ハ ダ゛キシメル
カコ ノ アメ ハ キレイダ゛
ショウネン ガ゛ ムナシク トキメク
カコ ノ オモウ ハナ ヨ
アオイ レモン ハ ワカイ
カガ゛ヤキ ノ ツキ ハ サビ゛シイ
カガ゛ヤキ ノヨウニ ショウジ゛ョ ハ アカルイ
```

もともと人工知能シミュレーションのつもりが日本語を作成するプログラムと言ったほうがいいようなプログラムになりました。できる人は解析して，拡張してみてください。文のパターンは，REM文でPAT～と書かれている部分で作られていて，大まかに８つに分かれています。パターンの内容は変数Ｌ＄中に，Ａ～Ｎの文字として扱われています。その文字が，順に，名詞，動詞，形容詞，形容動詞，副詞，が，は，だ，のようだ，のように，の，を，よ，の各品詞や助詞に対応しています。Ｌ＄＝“ＡGB”なら，（名詞）＋「は」＋（動詞），たとえば，「星は遊ぶ」などの文になるわけです。文の形はもっといろいろ考えられますので，パターンの拡張をしてみてください。その他，単語に0～9の，例えば楽しさや明るさのパラメータをつけて，暗い詩，明るい詩を作れるようにすれば，より人工知能に近くなるのではないでしょうか。ぜひ挑戦してほしいと思います。

リスト

```
1000 PRINT"*************************
1010 PRINT"* ｼｼﾞﾝ ﾆ ﾅﾘﾀｲ ver.1,1  *
1020 PRINT"*     by M.Kudoh       *
1030 PRINT"*************************
1040 DIM TA$(4,255),GY$(11),J$(7),MX(4)
1050 FOR I=0 TO 7 :READ J$(I):NEXT
1060 FOR I=0 TO 4
1070  J=0
1080  READ T$
1090  IF T$<>"*" THEN TA$(I,J)=T$:J=J+1:GOTO 1080
1100  MX(I)=J-1
1110 NEXT
1120 '
1130 PRINT:PRINT" =====<< MENU >>====="
1140 PRINT:PRINT"   1.   ｼ ｻｸｾｲ       "
1150 PRINT"   2.   ｵｼﾏｲ"
1160 K=1:PRINT:INPUT"ｽｳｼﾞ ｦ ﾆｮｳﾘｮｸ ｼﾃｸﾀﾞｻｲ ",K
1170 ON K GOTO 1240,1210
1180 GOTO 1130
1190 '
1200 '      END
1210 END
1220 '
1230 '    ｼ ｻｸｾｲ
1240 FOR I=0 TO 11:GY$(I)="":NEXT
1250 FOR LN=0 TO 11
1260  L$="":P=INT(RND(1)*8)+1
1270  ON P GOSUB 1440,1510,1580,1660,1720,1790,1840,1930
1280  GOSUB 1990
1290 NEXT LN
1300 '
1310 PRINT
1320 FOR I=0 TO 11
1330 PRINT RIGHT$(STR$(I+1),2);". ";GY$(I):NEXT
1340 PRINT:INPUT" Printer ﾆ ｼｭﾂﾘｮｸ ｼﾏｽｶ ? (Y or N)",K$
1350 IF (K$<>"Y")*(K$<>"y")*(K$<>"ﾝ") THEN GOTO 1130
1360 N=0:INPUT" ﾅﾝｷﾞｮｳﾒ ?  ｾﾞﾝﾌﾞ...Only [RET] ",N
1370 IF (N<0)+(N>12) THEN GOTO 1310
1380 I0=0:I1=11:IF N<>0 THEN I0=N-1:I1=N-1
1390 FOR I=I0 TO I1
1400   LPRINT GY$(I) ' PRINT/P GY$(I)
1410 NEXT:GOTO 1310
1420 '
1430 ' PAT 1
1440 N=INT(RND(1)*100)
1450 HI=ABS(69*(N<=5)+68*(N>5)*(N<=20)+67*(N>20)*(N<=60)+66*(N>60))
1460 L$=L$+CHR$(HI)+"A"
1470 IF INT(RND(1)*2)=0 THEN L$=L$+"M"
1480 RETURN
1490 '
1500 ' PAT 2
1510 L$=L$+"A"+CHR$(INT(RND(1)*2)+70)
1520 N=INT(RND(1)*10)
1530 HI=ABS(69*(N<=3)+68*(N>3)*(N<=6)+67*(N>6))
1540 L$=L$+CHR$(HI)+CHR$(INT(RND(1)*3)+65)
1550 RETURN
1560 '
1570 ' PAT 3
1580 L$=L$+CHR$(65,71)
1590 N=INT(RND(1)*100):IF N>=20 THEN 1610
1600 L$=L$+"AH":RETURN
1610 HI=ABS(68*(N>=20)*(N<=40)+67*(N>40)*(N<=70)+66*(N>70))
1620 L$=L$+CHR$(HI)
1630 RETURN
1640 '
1650 ' PAT 4
1660 GOSUB 1580
1670 IF RIGHT$(L$,1)="H" THEN L$=LEFT$(L$,LEN(L$)-1)+"K"
1680 L$=L$+"A"+CHR$(INT(RND(1)*2)+73)
1690 RETURN
1700 '
1710 ' PAT 5
1720 N=INT(RND(1)*10)
1730 HI=ABS(68*(N<=2)+66*(N>2)*(N<=5)+67*(N>5))
1740 L$=L$+CHR$(HI)
1750 GOSUB 1580
1760 RETURN
1770 '
1780 ' PAT 6
```

```
1790 L$=L$+"AJ"
1800 GOSUB 1580
1810 RETURN
1820 '
1830 ' PAT 7
1840 IF INT(RND(1)*2)=0 THEN 1870
1850 GOSUB 1580:IF MID$(L$,3,1)<>"B" THEN 1890
1860 GOTO 1880
1870 GOSUB 1510:IF MID$(L$,4,1)<>"B" THEN 1890
1880 L$=LEFT$(L$,2)+"AL"+RIGHT$(L$,LEN(L$)-2):RETURN
1890 L$="A"+CHR$(INT(RND(1)*2)+70)+"ALB"
1900 RETURN
1910 '
1920 ' PAT 8
1930 L$=L$+"AK"
1940 IF INT(RND(1)*2)=0 THEN GOSUB 1440 :RETURN
1950 GOSUB 1580
1960 RETURN
1970 '
1980 ' ｺﾄﾊﾞ ｴﾗﾋﾞ
1990 FOR I=1 TO LEN(L$)
2000  N=ASC(MID$(L$,I,1))-65
2010  IF N=-65 THEN 2210
2020  IF N>=5 THEN GY$(LN)=GY$(LN)+" "+J$(N-5):GOTO 2210
2030  ON N+1 GOTO 2050,2050,2080,2150,2050
2040 ' ｿﾉﾀ
2050  GOSUB 2240
2060  GOTO 2210
2070 ' ｹｲﾖｳｼ
2080  GOSUB 2240
2090  IF I=LEN(L$) THEN 2210
2100  H$=MID$(L$,I+1,1):K$="ｲ"
2110  IF (H$="B")+(H$="C") THEN K$="ｸ"
2120  GY$(LN)=LEFT$(GY$(LN),LEN(GY$(LN))-1)+K$
2130  GOTO 2210
2140 ' ｹｲﾖｳﾄﾞｳｼ
2150  GOSUB 2240
2160  IF I=LEN(L$) THEN 2210
2170  H$=MID$(L$,I+1,1):K$="ﾀﾞ"
2180  IF (H$="B")+(H$="C") THEN K$="ﾆ"
2190  IF H$="A" THEN K$="ﾅ"
2200  GY$(LN)=LEFT$(GY$(LN),LEN(GY$(LN))-2)+K$
2210 NEXT I
2220 RETURN
2230 '
2240  T=INT(RND(1)*(MX(N)+1))
2250  GY$(LN)=GY$(LN)+" "+TA$(N,T)
2260  RETURN
2270 '
2280 DATA ｶﾞ,ﾊ,ﾀﾞ,ﾉﾖｳﾀﾞ,ﾉﾖｳﾆ,ﾉ,ｦ,ﾖ
2290 '
10000 DATA ﾅﾐ,ﾊﾅ,ｿﾗ,ﾎｼ,ﾖｿﾞﾗ,ﾋﾄﾐ,ｶｺ,ﾅﾐﾀﾞ,ｱﾒ,ｶｶﾞﾔｷ,ﾎﾀﾙ,ｷﾞﾝｶﾞ,ｼｮｳﾈﾝ,ﾚﾓﾝ,ｵﾓｲﾃﾞ
10010 DATA ｼｮｳｼﾞｮ,ﾕｳﾋ,ﾊﾂｺｲ,ﾂｷ,ｾｰﾗｰﾌｸ,*
10020 DATA ﾅｸ,ﾜﾗｳ,ﾄｷﾒｸ,ｽｽﾑ,ﾌｯﾃｲﾙ,ﾓｴﾙ,ﾒｻﾞﾒﾙ,ﾈﾑﾙ,ｶｶﾞﾔｸ,ｵﾓｳ,ｶﾚﾀ,ﾄﾌﾞ,ﾋｷｻｸ,ﾅｶﾞﾚﾙ
10030 DATA ｱｿﾌﾞ,ｽｷﾞｻｯﾀ,ﾀﾞｷｼﾒﾙ,ﾀﾜﾑﾚﾙ,ﾃﾗｼﾃｲﾙ,ｼｯﾃｲﾙ,*
10040 DATA ﾑﾅｼｲ,ｳﾂｸｼｲ,ﾔｻｼｲ,ｸﾙｼｲ,ｻﾋﾞｼｲ,ｶﾅｼｲ,ﾀｶｲ,ｼﾛｲ,ﾏﾌﾞｼｲ,ﾂﾒﾀｲ,ｲﾄｼｲ,ｳﾙﾜｼｲ
10050 DATA ﾊｹﾞｼｲ,ﾅﾔﾏｼｲ,ｱｵｲ,ｸﾗｲ,ｺｲｼｲ,ｱｲﾀｲ,ｱｶﾙｲ,ﾜｶｲ,*
10060 DATA ｼｽﾞｶﾀﾞ,ｼｭｰﾙﾀﾞ,ｷﾚｲﾀﾞ,ｹﾞﾝｿｳﾃｷﾀﾞ,ｱﾜﾚﾀﾞ,ﾎﾉｶﾀﾞ,ｽｷﾀﾞ,ﾍｲﾜﾀﾞ,ﾐﾘｮｸﾃｷﾀﾞ
10070 DATA ﾜｶﾞﾏﾏﾀﾞ,ﾐﾜｸﾃｷﾀﾞ,ﾌｼｷﾞﾀﾞ,ｵﾀﾞﾔｶﾀﾞ,ﾏｯｶﾀﾞ,ﾁｯﾎﾟｹﾀﾞ,*
10080 DATA ｼﾄｼﾄ,ﾀﾏﾀﾏ,ｽｯｶﾘ,ｺｳｺｳﾄ,ﾄﾞｷﾄﾞｷ,ﾜｸﾜｸ,ﾕｯｸﾘ,ｵｵｲﾆ,ﾏｯﾀｸ,ﾓｳ,ｽﾃﾞﾆ,ﾄｷﾆﾊ,ｲﾂｶﾊ
10090 DATA ﾓｸﾓｸﾄ,ﾏｯﾀﾘﾄ,*
```

X1turbo, NEW BASICの場合の変更点

```
2100 H$=MID$(L$,I+1,1):K$="い"
2110 IF (H$="B")+(H$="C") THEN K$="く"
2120 GY$(LN)=LEFT$(GY$(LN),LEN(GY$(LN))-2)+K$
2170 H$=MID$(L$,I+1,1):K$="だ"
2180 IF (H$="B")+(H$="C") THEN K$="に"
2190 IF H$="A" THEN K$="な"
2280 DATA が,は,だ,のようだ,のように,の,を,よ
10000 DATA 波,花,空,星,夜空,瞳,過去,涙,雨,輝き,蛍,銀河,少年,レモン,思い出,少女
10010 DATA 夕陽,初恋,月,セーラー服,*
10020 DATA 泣く,笑う,ときめく,進む,降っている,燃える,目覚める,眠る,輝く,想う
10030 DATA 枯れた,飛ぶ,引き裂く,流れる,遊ぶ,過ぎ去った,抱き締める,戯れる
10040 DATA 照している,知っている,*
10050 DATA 虚しい,美しい,優しい,苦しい,淋しい,哀しい,高い,白い,まぶしい
10060 DATA 冷たい,愛しい,麗しい,激しい,悩ましい,蒼い,暗い,恋しい,逢いたい
10070 DATA 明るい,若い,*
10080 DATA 静かだ,シュールだ,奇麗だ,幻想的だ,哀れだ,ほのかだ,好きだ,平和だ
10090 DATA 魅力的だ,わがままだ,魅惑的だ,不思議だ,穏やかだ,真っ赤だ,ちっぽけだ,*
10100 DATA しとしと,たまたま,すっかり,こうこうと,ドキドキ,ワクワク,ゆっくり
10110 DATA 大いに,全く,もう,既に,時には,いつかは,黙々と,まったりと,*
```

X1,NEW BASICの場合の変更点

```
1310 CLS4:COLOR 5
1320 OPEN"O",#1,"GR:":FOR I=0 TO 11
1330 PRINT #1,RIGHT$(STR$(I+1),2);". ";GY$(I):NEXT:CLOSE
1340 COLOR7:LOCATE0,24:INPUT" Printer ﾆ ｼｭﾂﾘｮｸ ｼﾏｽｶ ? (Y or N)",K$
1350 IF (K$="Y")+(K$="y")+(K$="ﾝ") ELSE CLS4:GOTO 65535
1360 N=0:CLS:LOCATE0,24:INPUT" ﾅﾝｷﾞｮｳﾒ ?  ｾﾞﾝﾌﾞ...Only [RET] ",N
1370 IF (N<0)+(N>12) THEN CLS:GOTO 1340
1390 OPEN"O",#1,"LP:":FOR I=I0 TO I1
1400   PRINT #1,GY$(I)
1410 NEXT:CLOSE:CLS:GOTO 1340
```

MZ-1500バーコードリーダ応用術

MZ-1500の周辺アプリを考える会
Tatsumi Takashi
辰巳 卓

わーい！ またやってまいりました“MZ-1500の周辺アプリを考える会”。先月号では我等が会のボス，幸氏がMCRの接続に挑戦しましたが，まあなんと，MCRの値段の高いこと。噂によるとMCRを買われた幸氏は1カ月間，雲や霞を食べて生活したとか（そんなアホな！）。

冗談はさておき，実際のところマークカードのコーディングには大変苦労されたと思います（御苦労さん）。そこで今回，私はMCRに対抗できるのは何かと考えたあげく，バーコードリーダ（以下BCRと略）に取り組んでみる事にしました。バーコードは，最近ではかなりポピュラーになってきておりまして，お菓子の箱や缶ジュースの缶等に印刷されている縞模様のかたまりの事です。このバーコードをライトペンのようなもの（ライトペンではない）や，ドライヤーのような形をしたバーコードスキャナーと呼ばれる入力装置で読み取り，その品物が何であるかを認識し，ホストとなるコンピュータ等に知らせるのです。だいたい私たちがよく見かけるタバコやお菓子の箱に付いているバーコードはJAN(Japan Article Numbering）と呼ばれるコード体系のものがよく使われているようです。

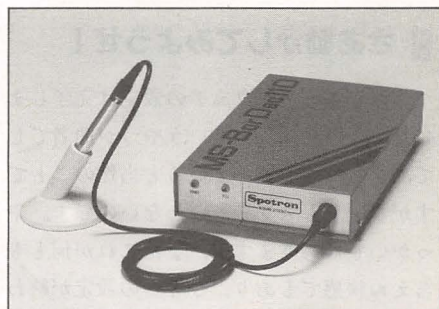
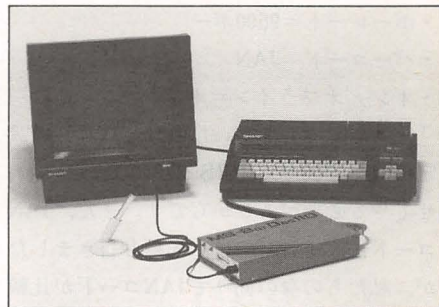
さて，BCRをMZ-1500に接続する事が決まったところで，どこかに適当なBCRがないかと日本全国を捜し回りました（ナント大袈裟な！）。BCRを捜し求めて3千里。なにしろMZの周辺機器の中にはBCRはないのですから。いろいろと捜したところ，ようやく㈱メカノシステムズという会社よりSpotron[TM]のBCR，MS-BarDec110（16万9千円）が発売されているという事が判明。早速，清水の舞台から飛び降りたつもりで大枚，16万9千円をはたいて購入しました（すごい出費だー）。また，読者の方より「先月号に続いてそんな高価な周辺機器を買いやがって，俺を殺すつもりか！」というお叱りの言葉が聞えてくるような気がしますが，そこはなんとかMZ-1500のためと思ってグッとこらえてください。実際のところ会のメンバーからは，毎日，白い目で見られているのですから。

このBCRは，RS-232Cで接続でき，出力コネクタもD-25S（25ピンDサブコネクタ）ですのでMZ-1500との接続にも相性が良く，また，読み取り可能なバーコードも6種類あり，なかなか高性能です。

さあ接続だ！

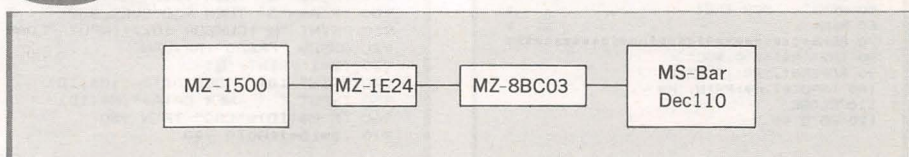
BCRが入手できたので，さっそくMZ-1500との接続に取りかかりましょう。今回のバーコードシステムに必要なものは以下のとおりです。

- MZ-1500＆ディスプレイ（当然デスネ）
- BCR（MS-BarDec110）
- MZ-1E24（低価格RS-232Cインタフェイス）
- MZ-8BC03（RS-232Cケーブル）
- MZ-1P17（高機能・ハイコストパフォーマンス サーマル漢字プリンタ）

MZ-1P17に関しては，バーコードプリンタとして使用しますので専用バーコードプリンタがあれば特に必要ありません。しかし，既にBCRで16万9千円もの投資を行っており，私のサイフの中はもうスッカラカン。したがって，これ以上の出費は取りも直さず身の破滅を招くことになりかねないので，ここはひとつソフトの力でMZ-1P17をみごとバーコードプリンターに変身させることにしましょう。

さて，以上のものが用意できたら，それぞれの設定に取りかかりましょう。まず，MZ-1E24からです。MZ-1E24の設定は以下のようにします。

- 使用チャンネル＝Aチャンネル
- 結線モード＝BIモード
- 端子モード＝ターミナルモード
- ボーレート＝9600ボー
- ポートアドレス＝B0H～B3H（工場出荷時設定）
- JB-M＝OFF

以上の設定を，説明書を見ながら確実に行

図1 接続図

ってください。ただし，チャンネルとボーレートは好みのものを使用しても構いません。次にBCR本体の設定です。こいつは，とりあえず以下のように設定します。

・終了コード＝CR（0DH）
・データビット＝8ビット
・パリティチェック＝DISABLE
・ストップビット＝2ビット
・ボーレート＝9600ボー
・バーコード＝JAN
・オン／オフライン＝オンライン

「とりあえず」と書いたのは，これらの設定はMZ-1500本体のRS-232Cの設定と合わせてやる必要があるからです。また，JANコードに設定したのは，先程も書きましたが，私たちの身の回りでJANコードが比較的簡単に見つけることができるからです。

さあ動かしてみようぜ！

これで各ハードウェアの設定は完了しました。この初期設定というやつが曲者でして，ひとつでも違っていると当然のことですがウンともスンともいわないのです。やっかいではありますが，またこれが何とも言えぬ快感でもあり，ひとつの設定が終わるごとに私のワクワクボルテージは上がっていくのです。ちょっと横道にそれてしまいましたネ。さて，おもむろにMZ-1500の内部スロットにMZ-1E24を差し込み，ケーブルをBCRに接続します。はやる心を静めながら電源ON！ 待て待て，慌てるでない。電源をONにはしてみたものの，動かすソフトがないではないか。「パソコンも，ソフトがなければただの箱」ということは賢明なMZファンの方々なら100年も前から知っておられるでしょう。ということで何か手短なソフトを用意しましょう。

大好きなBASICを起動し，リスト1を入力し，F1キー（RUN）を押して実行してください。実行と同時にバーコードの受信

リスト1 BCRテストプログラム

```
10 REM**********************************
20 REM*                                *
30 REM*  PROGRAM 1                     *
40 REM*                                *
50 REM*    BCR TEST                    *
60 REM*                                *
70 REM**********************************
80 INIT"RS1:$00,$8C"
90 ROPEN#1,"RS1:RS232C"
100 INPUT#1,M$:PRINT M$
110 CLOSE
120 GOTO 90
```

リスト2 BCR活用プログラム

```
10 REM**********************************
20 REM*                                *
30 REM*                                *
40 REM*  PROGRAM2                      *
50 REM*                                *
60 REM*   BAR CODE APLICATION          *
70 REM*                                *
80 REM*                                *
90 REM**********************************
100 DIM M$(100),DA$(30),DN$(10),DK$(20)
110 ON ERROR GOTO "ERROR"
120 RESTORE 1680:FOR I=0 TO 9:READ DN$(I):NEXT I
130 RESTORE 1690:FOR I=1 TO 26:READ DA$(I):NEXT I
140 RESTORE 1720:FOR I=0 TO 15:READ DK$(I):NEXT I
150 RESTORE 1730:READ DK$(20)
160 LABEL "MAIN":PRINT "▣"
170 CURSOR 10,5:PRINT "MUSIC PLAY---------1"
180 CURSOR 10,7:PRINT "MUSIC SAVE---------2"
190 CURSOR 10,9:PRINT "BAR CODE PRINT----3"
200 CURSOR 10,11:PRINT "BAR CODE READ ----4"
210 CURSOR 10,13:INPUT "      PUSH KEY (+CR)   ";A$
220 IF A$="1" THEN "MPLAY"
230 IF A$="2" THEN "MSAVE"
240 IF A$="3" THEN "BCPRINT"
250 IF A$="4" THEN "BCREAD" ELSE 210
260 REM
270 REM-----MUSIC PLAYING-----
280 REM
290 LABEL "MPLAY"
300 GOSUB "MENU"
310 IF DL<=1 THEN "MAIN"
320 FOR ID=1 TO DL-1:MUSIC M$(ID):NEXT ID
330 PRINT "▣":CURSOR 10,11:PRINT "MUSIC END":WAIT 1000:GOTO "MAIN"
340 REM
350 REM-----MUSIC DATA SAVE TO QD-----
360 REM
370 LABEL "MSAVE"
380 GOSUB "MENU"
390 IF DL=0 THEN "MAIN"
400 PRINT "▣":CURSOR 10,7:INPUT "SAVE DATA FILE NAME= ";F$
410 WOPEN #1,"QD:"+F$
420 FOR ID=1 TO DL:PRINT #1,M$(ID):NEXT ID
430 CLOSE
440 PRINT "▣":CURSOR 10,11:PRINT "MUSIC DATA SAVE COMPLETED":WAIT 1000:GOTO "MAI
N"
450 REM
460 REM-----BAR CODE PRINTING-----
470 REM
480 LABEL "BCPRINT"
490 GOSUB "MENU"
500 IF DL=0 THEN "MAIN"
510 FOR ID=1 TO DL
520 REM-----CONVERT # TO %-----
530 MO$="":ML=LEN(M$(ID))
540 FOR JD=1 TO ML
550 MK$=MID$(M$(ID),JD,1)
560 IF MK$="#" THEN MK$="%"
570 MO$=MO$+MK$
580 NEXT JD
590 MM$=MO$:GOSUB "BARPRNT"
600 NEXT ID
610 PRINT "▣":CURSOR 10,11:PRINT "BAR CODE PRINT COMPLETED":GOTO "MAIN"
620 REM
630 REM-----BAR CODE READ-----
640 REM
650 LABEL "BCREAD"
660 PRINT "▣":CURSOR 10,11:PRINT "RECEIVING BAR CODE"
670 INIT"RS1:$00,$8C":REM-----RS-232C INITIALIZE-----
680 ROPEN#1,"RS1:RS232C"
690 FOR ID=1 TO 100
700 INPUT#1,M$(ID):MO$="":ML=LEN(M$(ID))
710 FOR JD=1 TO ML
720 MK$=MID$(M$(ID),JD,1)
730 IF MK$="%" THEN MK$="#"
740 MO$=MO$+MK$
750 NEXT JD
760 M$(ID)=MO$
770 CURSOR 10,13:FOR IS=1 TO 39:PRINT CHR$($20);:NEXT IS:CURSOR 10,13:PRINT M$(I
D)
780 IF M$(ID)="END" THEN 800
790 NEXT ID
800 DL=ID:CLOSE:GOTO "MAIN"
810 REM
820 LABEL "MENU"
830 PRINT "▣"
840 CURSOR 10,5:PRINT "DATA ALREADY EXIST--1"
850 CURSOR 10,7:PRINT "DATA LOAD FROM QD---2"
860 CURSOR 10,9:PRINT "DATA MANUAL INPUT---3"
870 CURSOR 10,11:INPUT "      PUSH KEY (+CR)   ";A$
880 IF A$="1" THEN RETURN
890 IF A$="2" THEN 910
900 IF A$="3" THEN 930 ELES 800
910 PRINT "▣":CURSOR 10,7:INPUT "LOAD DATA FILE NAME= ";F$
920 GOSUB "FREAD":RETURN
930 ID=1:PRINT "▣"
940 PRINT ID;"  OLD DATA=";M$(ID)
950 INPUT "    NEW DATA=";M$(ID)
960 IF M$(ID)="END" THEN 980
970 ID=ID+1:GOTO 940
```

待ちとなりますから，適当にバーコードを見つけてきて（とは言ってもJANコードのものでないと困りますが），バーコードスキャナーを持ってバーコードの上をこすってみてください。「やった！」ピッ！という音とともにCRT画面上に受信データが出力されたでしょう。GOTO文でループになっていますので何度でもリトライ可能です。まあしばらくはお菓子の箱でも持ってきて遊んでみてください。

いよいよ本番だ！真打登場

これでBCRは動くようになりました。でもこれで「めでたし，めでたし」とペンを置いたのでは，MZ-1500の周辺アプリを考える会々員としては，由々しき問題であり，悪くすれば会を破門になる可能性も十分にあると考え，次のようなアプリケーションを作ることにしました。

1. MZ-1P17をバーコードプリンタとして使う。
2. 音楽データをバーコードで入力する。
3. 音楽データをバーコードでプリントアウトする。

以上のコンセプトの順に進んでいくことにします。

MZ-1500BASICの音楽データには，アルファベットが必要となりますのでバーコードはCODE-39を使用します。先程まではずっとJANのコードでやってきましたので，BCRのコード設定をCODE-39に変更してください。また，“#”記号がシャープとして必要ですがCODE-39にはありませんので，その代わりに“%”のバーコードを使用します。ですからプログラム中でコード“#”↔“%”の変換が必要です。

少し前おきが長くなりましたが，リスト2をこつこつ入力していってください。入力が終わったらトラブルが起こる前にQDにセーブしておきましょう。このプログラムも最初はもっと短かったのですが，作っている最中に，私の回りにいる人間たち（当然，うちの会員連中です）が，「ここはこうした方がええで」とか，「こんな機能もあった方がええんとちゃうか（良いのではないか）」と完全な大阪弁でちょっかいを出してくるのです。そう言われるとギャラリー受けを狙らう私といたしましては，希望に沿うように努力する次第でございまして，

```
980 DL=ID:RETURN
990 LABEL "FREAD"
1000 ROPEN #1,"QD:"+F$
1010 FOR ID=1 TO 100
1020 INPUT#1,M$(ID)
1030 IF M$(ID)="END" THEN 1050
1040 NEXT ID
1050 CLOSE:DL=ID:RETURN
1060 REM
1070 REM*********************************
1080 REM*                               *
1090 REM* BAR CODE PRINT   BY MZ-1P17 *
1100 REM*         MM$--DATA            *
1110 REM*                               *
1120 REM*********************************
1130 LABEL "BARPRNT"
1140 REM
1150 INIT"LPT:S5,$0D":REM-----PRINTER INITIALIZE-----
1160 PRINT/P CHR$($A):PRINT/P:PRINT/P:PRINT/P CHR$($1B);"%6";CHR$(14);
1170 REM
1180 FOR IP=1 TO 3
1190 REM
1200 REM-----START CHARACTER (*) PRINT-----
1210 REM
1220 BA$=DK$(20):GOSUB "BPRINT"
1230 REM
1240 REM-----DATA CHARACTER PRINT-----
1250 REM
1260 ML=LEN(MM$):REM-----ML=DATA  LENGTH-----
1270 FOR II=1 TO ML
1280 ST$=MID$(MM$,II,1):S1=ASC(ST$):BA$=DA$(24)
1290 IF S1>=$20 AND S1<=$2F THEN BA$=DK$(S1-$20)
1300 IF S1>=$30 AND S1<=$39 THEN BA$=DN$(S1-$30)
1310 IF S1>=$41 AND S1<=$5A THEN BA$=DA$(S1-$40)
1320 GOSUB "BPRINT"
1340 NEXT II
1350 REM
1360 REM-----STOP CHARACTER (*) PRINT-----
1370 REM
1380 BA$=DK$(20):GOSUB "BPRINT"
1390 PRINT/P
1400 NEXT IP
1410 PRINT/P CHR$($A);
1420 RETURN
1430 REM
1440 REM-----PRINT SUBROUTINE-----
1450 REM
1460 LABEL "BPRINT"
1470 PRINT/P CHR$($1B);"%1";CHR$(0,36);
1480 FOR I=1 TO 9
1490 B1$=MID$(BA$,I,1)
1500 IF B1$="0" THEN PRINT/P "πππππππππ";
1510 IF B1$="1" THEN PRINT/P "ππππππππππππππππππ";
1520 IF B1$="2" THEN PRINT/P CHR$(0,0,0,0,0,0,0,0,0);
1530 IF B1$="3" THEN PRINT/P CHR$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
1540 NEXT I
1550 REM
1560 REM-----CHARACTER SPACE PRINT-----
1570 REM
1580 PRINT/P CHR$($1B);"%1";CHR$(0,6);CHR$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
1590 RETURN
1600 END
1610 REM
1620 REM  CODE-39 BAR CODE DATA
1630 REM         0--NARROW BAR
1640 REM         1--WIDE   BAR
1650 REM         2--NARROW SPACE
1660 REM         3--WIDE   SPACE
1670 REM
1680 DATA "020312120","120302021","021302021","121302020","020312021","120312020
","021312020","020302121","120302120","021302120"
1690 DATA "120203021","021203021","121203020","020213021","120213020","021213020
","020203121","120203120","021203120","020213120"
1700 DATA "120202031","021202031","121202030","020212031","120212030","021212030
","020202131","120202130","021202130","020212130"
1710 DATA "130202021","031202021","131202020","030212021","130212020","031212020
"
1720 DATA "021202120","DM","DM","DM","030303020","020303030","DM","DM","DM","DM"
,"030212120","030203030","DM","030202121","130202120","030302030"
1730 DATA "030212120"
1740 END
1750 LABEL "ERROR"
1760 PRINT "■":CURSOR 10,5
1770 IF ERL=320 THEN PRINT "ERROR !!!---ILLEGAL MUSIC DATA"
1780 IF ERL>380 AND ERL<440 THEN PRINT "ERROR !!!---MUSIC DATA SAVE"
1790 WAIT 1000
1800 RESUME 160
```

とうとうこのようなプログラムになってしまいました。

また，自分が個人として使うのであれば，このようにREM文を入れることもないのですが，いざ人様に見ていただくとなると，とたんに様子が変わってしまい，少しでもカッコ良く，また少しでも理解しやすくと，ない知恵をふり絞って考えましたが，やはりプログラムの中身を見られるというのは何か自分の通知票でも見られているようで，気恥ずかしいものです。

話がかなりプログラムの本題とかけは離れてしまいました。ここらあたりで再び軌道修正とまいりましょう。こつこつと入力

したリスト2を，とにかく実行させてみましょう。メインメニューがCRT画面にこのように表示されます。

MUSIC PLAY……………1
MUSIC SAVE……………2
BAR CODE PRINT……3
BAR CODE READ……4
PUSH KEY（+CR）

と表示されます。1は音楽の演奏，2は音楽データのQDへの登録，3は音楽データをバーコードでプリント，4はバーコードでプリントされた音楽データを読み込めるようになっています。さらに1・2・3のうちいずれかを選択すると，サブメニューがCRT画面上に表示されます。

DATA ALREADY EXIST…1
DATA LOAD FROM QD……2
DATA MANUAL INPUT……3
PUSH KEY（+CR）

ここで，1はすでにデータがMZ-1500内にある場合に，キーボードより入力します。ですからこのプログラムを実行し，メインメニューで1か2か3を選んだ直後に，サブメニューで1を入力すると，まだMZ-1500内に音楽データが無いために，メインメニューに戻ってしまいます。2は音楽データをQDからロードする場合に使用します。当然のことながらこの場合にはQD内に音楽データがBSDファイルで登録されていることが必要です。またこのデータの最後は，“END”という文字データであることが必要です。データのロードに際しては，ファイルネームが聞かれますので正しく入力してください。

最後に3ですが，これはキーボードよりマニュアルで音楽データを入力する場合に使います。3を入力すると，画面には，

DATA=

と表示されますので，BASICに従って音楽データを入力し，CRキーを入力してください。ただし，この場合の注意として，

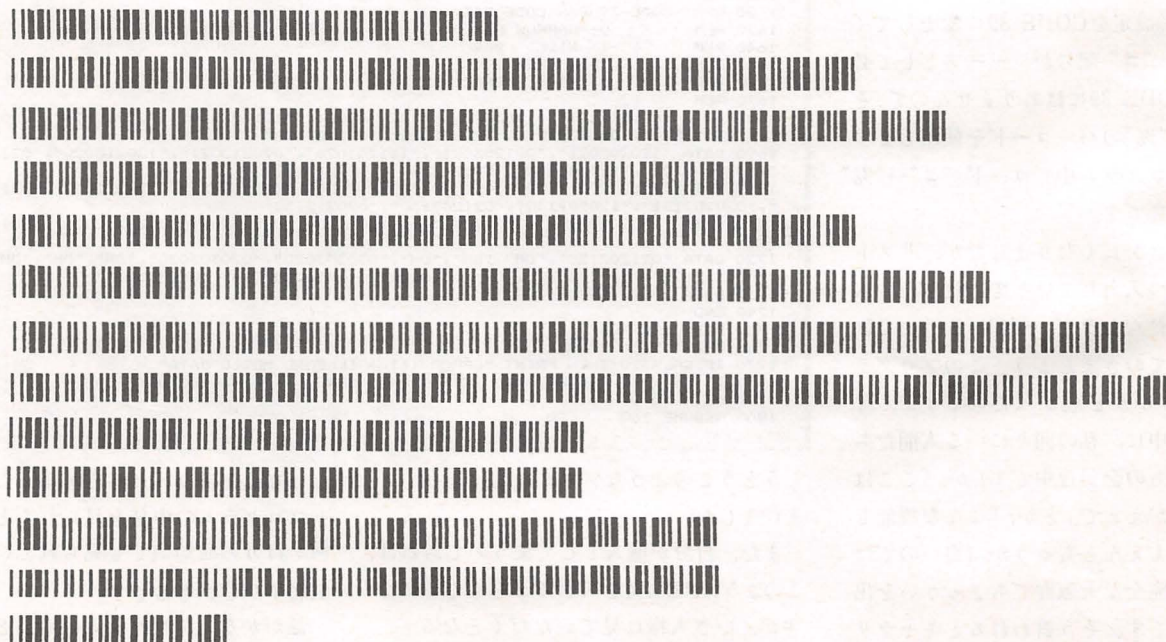
1）CRキーを入力するまでのひとつのデータの長さは，1～30キャラクタまでとします（BCRの読み取り能力の関係上）。また，31キャラクタ以上になるときは，データをいくつかの部分に分割して入力してください。
2）最後のデータは，
DATA=END ⏎（CR）
で終わるようにしてください。このデータの入力によってメインルーチンへ戻ることができます。

さて，バーコードのプリントですが，これは，MZ-1P17のビットイメージコマンドを使用します。BCRの読み取り精度を上げるため，バーコードにはある程度の濃さが必要となってきますので，同じデータを続けて3回印字します。ただし先にプリントされたデータとうまく重なるように改行量は調節してあります。CODE-39の各キャラクタのワイドエレメント（ワイドバーおよびワイドスペース）には，ビットイメージで6カラム分を，またナローエレメント（ナローバーおよびナロースペース）には，3カラム分を使用します。読み取り精度を上げるためには，インクリボンを使用するよりも感熱紙を使用した方がベターだと思います。以上が今回のBCRプログラムの簡単な説明です。

プログラムはBASICしか使用していない（マシン語なんてもっての他）ので，特に難解という事はないと思います。エラー処

リスト3 六甲おろし

```
10 DATA  "T7V15M8S0"
20 DATA  "G7E5G5+C3+C5+C3G7"
30 DATA  "A5B3+C3+D5+C3A3G8R5"
40 DATA  "G7A5B5+C6+D3+E7"
50 DATA  "+D5+C5B5A3G3+C8R5"
60 DATA  "+D6+D3+D5G5+C5+E5+G7"
70 DATA  "+E5+G5+F3+E3+D3+C3+D8R5"
80 DATA  "+G6+G3+E5+C3B3A5B5+C5+D5"
90 DATA  "+E8+D5+C8R5"
100 DATA  "G7+C7+C5+E8"
110 DATA  "+G7+G7+G3+G7R6"
120 DATA  "+E7+D5+D5+C8R5"
130 DATA  "END"
```

六甲おろし（バーコードリーダで読めます）

理や入力方法，表示方法などまだまだ工夫できるところがあると思いますが，それらは賢明なる読者諸兄に委ねることにします。サンプル音楽として今シーズン日本一になりました阪神タイガースの「六甲おろし」のデータを載せておきますので参考にしてください。

とりあえずBCRの変わった使い方のひとつを考えてみましたが，もっと他におもしろい使い方もあると思います。たとえば，プログラムのリストをバーコード化するとか，サークルやクラブ等で暗号を作っておき，そのバーコードを読まないとシステムが起動しないとか，また，個人のIdentityコードを決めるとかです。あとは，皆さんで考えてみてください。

今回使用したBCRはいかがでしたでしょうか。今日の技術進歩は目ざましいものがあり，BCRの使い方にも予想外のものが出てくるかもしれませんね。

さて，この原稿を書いている間に，我等が会にひとつの荷物が届けられました。中を開けてみると何かロボットらしきものが入っているじゃありませんか。いずれこの誌面にも登場するのではないかと楽しみにしています。では，再び私とこの誌面でお会いする日まで，さらばじゃ。とあいさつをひととおりすませたあとで，ナント，バーコードの基本的解説が続くんですね。これが，しつこいなんて言わないで，きっと役に立つから読んでみてください。

各種バーコードの構成

これから各種バーコードの構成について簡単に説明します。なお，これらの内容については今回使用しましたMS-BarDec110の取扱説明書を参考にしています。

1．バーコードの構成

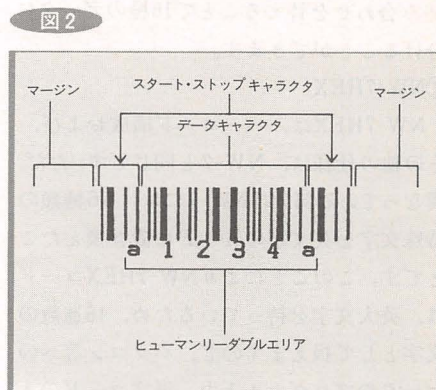
ほとんどのバーコードは図2のような構成になっています。

①スタート/ストップキャラクタ

これらのキャラクタは，それぞれバーコードシンボルの始まりと終わりを示すもので，スキャナに情報の入力方向やバーコードの種類を示します。バーコードはどちらの方向からもスキャニングできるようになっています。

②マージン（空白ゾーン）

スタート／ストップキャラクタに隣接する何も印刷しない空白域です。このマージンと，バーとスペースによる一定のパターンで，スキャナはそれが正しいコードであるかどうかを判断します。このマージンの幅はコード体系により規格が定められており，読み取り機によって若干の差はありますが，通常細いバーの横幅か，狭い方のスペースの幅の10倍以上は必要となります。このマージン幅が狭すぎると，スキャナはコードの確認ができません。

図2

③ヒューマンリーダブルエリア

ここに人間の判読可能な情報が文字で印刷されます。

2．コードの種類と特徴

①NW-7

NW-7は各キャラクタが7ビットのエレメント（7本のパターン）で表わされ，4本のバー（黒い部分）とその間の3本のスペースで構成されます。そのうちの2本，または3本はワイドエレメントで，他はナローエレメントです。また各キャラクタ間のスペースは，キャラクタギャップと呼び，ワイドスペースと同じ幅になっています。NW-7で表現できるキャラクタは，“0～9”までの数字と6種類の特殊文字(－＄：／．＋)，スタート／ストップコード(ａｂｃｄ)の計20種類です。

・スタート／ストップコード

スタート／ストップコード(a～b)は，データ列の最初と最後に必ず1文字ずつ付ける必要があります。また，その組み合わせは自由に使えるので，同じ“1234”というデータに対しても，スタート／ストップコードの

図3

a	B1	S1	B2	S2	B3	S3	B4

NB NS WS WS NB WS NB

ナロー：ワイド＝1：2.5～3.5
※NW-7の例

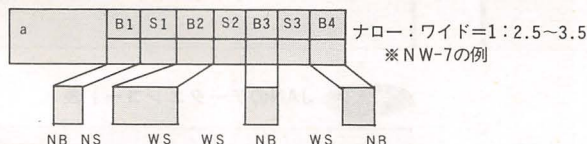



表1　NW 7コードのパターン構成表

CHARACTER	B1	S1	B2	S2	B3	S3	B4	CHARACTER	B1	S1	B2	S2	B3	S3	B4
	BARCORD PATTERN								BARCORD PATTERN						
0	0	0	0	0	0	1	1	－	0	0	0	1	1	0	0
1	0	0	0	0	1	1	0	$	0	0	1	1	0	0	0
2	0	0	0	1	0	0	1	.	1	0	1	0	1	0	0
3	1	1	0	0	0	0	0	／	1	0	1	0	0	0	1
4	0	0	1	0	0	1	0	：	1	0	0	0	1	0	1
5	1	0	0	0	0	1	0	＋	0	0	1	0	1	0	1
6	0	1	0	0	0	0	1	a	0	0	1	1	0	1	0
7	0	1	0	0	1	0	0	b	0	1	0	1	0	0	1
8	0	1	1	0	0	0	0	c	0	0	0	1	0	1	1
9	1	0	0	1	0	0	0	d	0	0	0	1	1	1	0

上の表で1はワイドエレメントで0はナローエレメントです。ここで，表を参照してデータ“a”に関するエンコード（パターン作成）したものを示します。

表2　NW-7HEXコードのパターン構成表

CHARACTER	B1	S1	B2	S2	B3	S3	B4	CHARACTER	B1	S1	B2	S2	B3	S3	B4
	BARCORD PATTERN								BARCORD PATTERN						
0	0	0	0	0	0	1	1	A	0	0	0	1	1	0	0
1	0	0	0	0	1	1	0	B	0	0	1	1	0	0	0
2	0	0	0	1	0	0	1	C	1	0	1	0	1	0	0
3	1	1	0	0	0	0	0	D	1	0	1	0	0	0	1
4	0	0	1	0	0	1	0	E	1	0	0	0	1	0	1
5	1	0	0	0	0	1	0	F	0	0	1	0	1	0	1
6	0	1	0	0	0	0	1	a	0	0	1	1	0	1	0
7	0	1	0	0	1	0	0	b	0	1	0	1	0	0	1
8	0	1	1	0	0	0	0	c	0	0	0	1	0	1	1
9	1	0	0	1	0	0	0	d	0	0	0	1	1	1	0

組み合わせを替えることで16種のデータに分けることができます。

②NW-7HEX

NW-7HEXは，バーコード構成および，その他の仕様は，NW-7と同じです。ただ，異なっている部分はNW-7における6種類の特殊文字を英文字のA～Fに置き換えたことです。このことによりNW-7HEXコードは，英大文字を持っているため，16進数の文字として扱えますので，パソコン等へのデータやプログラム入力，漢字コードの入力等に使用できます。

③CODE-39

CODE-39は各キャラクタが9ビットのエレメント（5本のバーと4本のスペース）で構成されます。9ビットのエレメントのうちの3本はワイドエレメントで，6本がナローエレメントです。また，キャラクタ間のスペースは，ワイドスペースと同じ幅

表3　CODE-39の構成

文字	バー	スペース	数値	文字	バー	スペース	数値
0	0 0 1 1 0	0 1 0 0	0	M	1 1 0 0 0	0 0 0 1	22
1	1 0 0 0 1	0 1 0 0	1	N	0 0 1 0 1	0 0 0 1	23
2	0 1 0 0 1	0 1 0 0	2	O	1 0 1 0 0	0 0 0 1	24
3	1 1 0 0 0	0 1 0 0	3	P	0 1 1 0 0	0 0 0 1	25
4	0 0 1 0 1	0 1 0 0	4	Q	0 0 0 1 1	0 0 0 1	26
5	1 0 1 0 0	0 1 0 0	5	R	1 0 0 1 0	0 0 0 1	27
6	0 1 1 0 0	0 1 0 0	6	S	0 1 0 1 0	0 0 0 1	28
7	0 0 0 1 1	0 1 0 0	7	T	0 0 1 1 0	0 0 0 1	29
8	1 0 0 1 0	0 1 0 0	8	U	1 0 0 0 1	1 0 0 0	30
9	0 1 0 1 0	0 1 0 0	9	V	0 1 0 0 1	1 0 0 0	31
A	1 0 0 0 1	0 0 1 0	10	W	1 1 0 0 0	1 0 0 0	32
B	0 1 0 0 1	0 0 1 0	11	X	0 0 1 0 1	1 0 0 0	33
C	1 1 0 0 0	0 0 1 0	12	Y	1 0 1 0 0	1 0 0 0	34
D	0 0 1 0 1	0 0 1 0	13	Z	0 1 1 0 0	1 0 0 0	35
E	1 0 1 0 0	0 0 1 0	14	—	0 0 0 1 1	1 0 0 0	36
F	0 1 1 0 0	0 0 1 0	15	・	1 0 0 1 0	1 0 0 0	37
G	0 0 0 1 1	0 0 1 0	16	SPACE	0 1 0 1 0	1 0 0 0	38
H	1 0 0 1 0	0 0 1 0	17	$	0 0 0 0 0	1 1 1 0	39
I	0 1 0 1 0	0 0 1 0	18	／	0 0 0 0 0	1 1 0 1	40
J	0 0 1 1 0	0 0 1 0	19	＋	0 0 0 0 0	1 0 1 1	41
K	1 0 0 0 1	0 0 0 1	20	%	0 0 0 0 0	0 1 1 1	42
L	0 1 0 0 1	0 0 0 1	21	*	0 0 1 1 0	1 0 0 0	

（注）SPACEはリーダーのデータ出力としては空白として出るので不読のデータではないことに注意してください。

図4-1　JAN標準

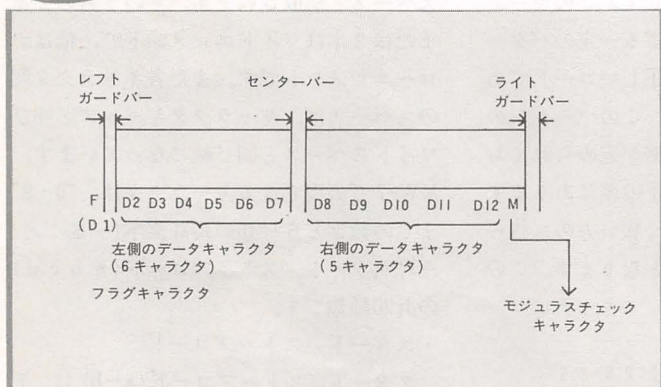



図4-2　JAN短縮

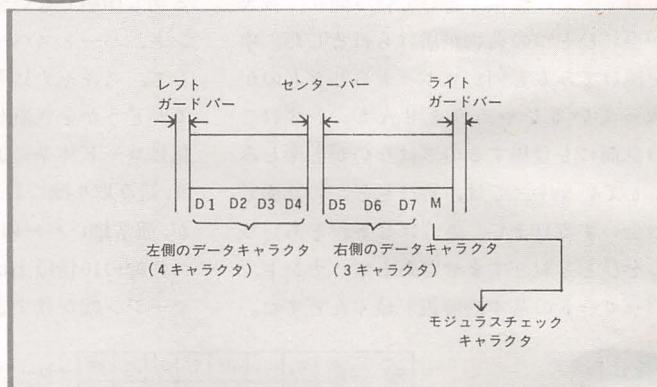



表4　フラグキャラクタによる左側データの構成

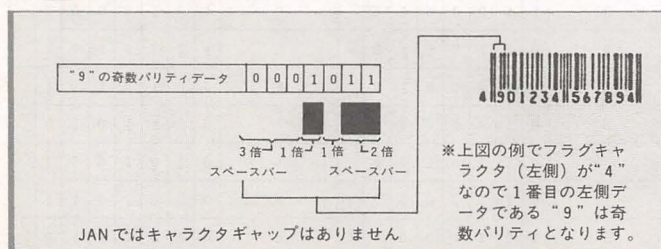
フラグキャラクタ	左側のデータキャラクタの組み合わせ					
F（D1）	D2	D3	D4	D5	D6	D7
0	O	O	O	O	O	O
1	O	O	E	O	E	E
2	O	O	E	E	O	E
4	O	E	O	O	E	E
5	O	E	E	O	O	E
6	O	E	E	E	O	O
7	O	E	O	E	O	E
8	O	E	O	E	E	O
9	O	E	E	O	E	O

＊Oは奇数パリティ
Eは偶数パリティ
を示します。

図5　JAN標準のエンコード例

表5　JANのデータエンコード表

10進数	左側のデータキャラクタ		右側のキャラクタ及びモジュラスチェックキャラクタ
	奇数パリティ	偶数パリティ	
0	0 0 0 1 1 0 1	0 1 0 0 1 1 1	1 1 1 0 0 1 0
1	0 0 1 1 0 0 1	0 1 1 0 0 1 1	1 1 0 0 1 1 0
2	0 0 1 0 0 1 1	0 0 1 1 0 1 1	1 1 0 1 1 0 0
3	0 1 0 0 0 1 1	0 1 0 0 0 0 1	1 0 0 0 0 1 0
4	0 1 0 0 0 1 1	0 0 1 1 1 0 1	1 0 1 1 1 0 0
5	0 1 1 0 0 0 1	0 1 1 1 0 0 1	1 0 0 1 1 1 0
6	0 1 0 1 1 1 1	0 0 0 0 1 0 1	1 0 1 0 0 0 0
7	0 1 1 1 0 1 1	0 0 1 0 0 0 1	1 0 0 0 1 0 0
8	0 1 1 0 1 1 1	0 0 0 1 0 0 1	1 0 0 1 0 0 0
9	0 0 0 1 0 1 1	0 0 1 0 1 1 1	1 1 1 0 1 0 0
レフトガードバー及びライトガードバー			1 0 1
センターバー			0 1 0 1 0

上記の表で“1”はバーで“0”はスペースを表しています。図5に表5を参照してJAN標準の左側データ“9”の奇数パリティについてエンコード(パターン作成)したものを示します。

になっています。CODE-39で表現できるキャラクタは36種類の英数文字（0～9，A～Z）と，7種類の記号文字（−，・，$，／，+，%，SPACE）です。

・スタート／ストップキャラクタ

CODE-39のスタートスタート／ストップキャラクタは“*”です。データ文字の最初と最後に必ず1文字つけてください。

・チェックキャラクタ

CODE-39は，データの信頼性を高めるためのチェックキャラクタが使用できます。このチェックキャラクタは，モジュラス43と呼ばれ次式より算出します。

例）“ABC”のモジュラスチェックキャラクタ。

表3の“A”，“B”，“C”の各データの“数値”の欄にある10，11，12を加えて，その結果を43で割ったときのあまりの値がモジュラスチェックキャラクタの“数値”になります。

“ABC” MOD43＝(10＋11＋12)
MOD43＝33

したがってチェックキャラクタは“X”になります。その結果，チェックキャラクタを付加したデータは“ABCX”となります。

④JAN

JANには，データ数が13あるJAN標準（図4-1）とデータ数が8のJAN短縮（図4-2）があります。このコードの特長は，バー／スペースの幅が4種類あることです。また，表現できるキャラクタは数字0～9のみです。

JAN標準にはフラグキャラクタが存在します（JAN短縮には存在しません）。フラグキャラクタは，バーコードの左側6データ（D2～D7）のバーコードイメージ（奇数パリティのデータにするか偶数パリティのデータにするか）を決定します。表4におけるフラグキャラクタに対するデータキャラクタの組み合わせて，奇数パリティ（0＝Odd）のコードを使うか，偶数パリティ（E＝Even）のコードを使うかを決め，表5からそのデータを決定します。JAN短縮の左側データは常に奇数パリティです。

・モジュラスチェックキャラクタ

これは，JANコードの最後に示されているデータで読み取り精度の向上に用いられます。

なお，この他のバーコードや，さらに詳しい内容については，専門の本におまかせします。それでは，これからほんとうの最後のごあいさつです。また，次回の企画を楽しみにしていてください。さらばじゃ。

〈BCR　問い合わせ先〉

販売代理店　コロンビア貿易株式会社
東京都千代田区猿楽町1-2-1
☎03(291)0641(代)
大阪市北区万才町4-12
☎06(361)7802

図6　モジュラスチェックキャラクタの求め方

まず図4-1におけるデータD1～D12を“490123456789”に対応させます。

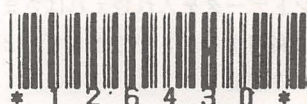
F(D1)	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
	9		1		3		5		7		9
4		0		2		4		6		8	

1：まず，2番目のキャラクタ（D2）からはじめて，すべての偶数番キャラクタ（上段）の値の和を求めると
D2＋D4＋D4＋D6＋D8＋D10＋D12＝9＋1＋3＋5＋7＋9＝34
2：そして1：で求められた値を3倍とすると　34×3102となり，これをDEとおきます（EVENDATA）
3：次に1番目のキャラクタ（D1）より，すべての奇数番キャラクタ（下段）の値の和を求めますと
D1＋D3＋D5＋D7＋D9＋D11＝4＋0＋2＋4＋6＋8＝24　DO（ODD）DATA
4：さらに，2：3：の結果の和を求めると　DE＋DO＝102＋24＝126
5：最後に，4：で求められた値より大きな10の倍数（この場合は130）との差を求めると，モジュラスチェックキャラクタが求められます。
M＝130−126＝4
これが求められたモジュラスチェック キャラクタでサンプルでの最後のキャラクタ“4”に一致していることが確かめられます。
短縮コードの場合はD1からD5を0で埋めD0～D12に7桁を入れて算出します。

表6　サンプルバーコード

このバーコードプリントはすべてシャープ製 BW-150S 高速バーコードライターでプリントしたものです。

1）NW-7（2 of 7）

2）CODE-39(3059)

3）JAN標準（UPC/EAN）

4）JAN短縮

サーキットエディタ

The部品箱

Sato Nobuo
佐藤 信夫

MZ-1500で回路図などの図面を作成するグラフィックエディタができました。この“The部品箱”はPCGを利用しているのでデータを入れ替えればさまざまな用途に使えます。プリントアウトはもちろん，画面スクロールやレイアウト表示も可能な便利なツールです。

このプログラムは，回路図などを書くためのエディタで，PCGによって登録した部品を画面上で組み合わせて図面を作成するというものです。みなさんはテンプレートというものをご存じでしょうか。いろいろな大きさの円や四角形などがくり抜かれた定規のようなもので，フローチャートを書くためのものや，実験器具の形に彫られたもの，住宅のテーブルやキッチン，トイレマークの入ったものまで種々のものがあります。ようするに既製の形のものは部品化して利用したいというわけです。回路図でも抵抗やコンデンサなどはパーツデータとして用意しておいたほうがいいだろうというのがこのプログラムの基本です。ここではPCGのフォントデータとして回路図用のパーツが登録されたものを掲載しますが，MZ-1500に同梱されているPCGエディタを使ってフォントデータを入れ替えればさまざまな用途に利用することができるでしょう。エディタの機能としては，種々のラインやボックス，消しゴムのほか，コピー機能や全体のレイアウトを見るマップ機能もあり，もちろん文字も書き込めます。画面はスクロールし，図面全体をプリントアウトすることが可能です。

図1　SUB.OBJの内容

アドレス	ラベル	働き
$C600	SETCHA	(X，Y)にキャラクタ No. でマップに書く
$C603	READCHA	(X，Y)の位置のキャラクタ No. を読む
$C606	LOADMAP	マップデータを QD から読む
$C609	SAVEMAP	マップデータを QD に書く
$C60C	MAPDESP	マップの一部を表示
$C60F	KEIIN	キー入力
$C612	MAPPRINT	レイアウト表示
$C615	MIHON	パーツを書くときの見本を書く
$C618	ALLCLS	マップをすべて消す

入力方法

このプログラムは4つの部分から成り立っています。

●ブヒン バコ（リスト1）

ここでは単にマシン語ルーチンを読み込んでいます。セーブは，

SAVE“ブヒン　バコ　V2.1A”

としてください。

●MAIN.BTX（リスト2）

いわずと知れたメインルーチンです。

SAVE“MAIN.BTX”

とします。

●FONT.OBJ（リスト3）

PCGの定義をするプログラムです。MZ-1500に付属のPCGエディタで作ったデータをG-RAMに転送するようになっています。

LIMIT $AFFF：BYE

を実行後，Mコマンドでリスト2を入力し，

＊SB000 CF51 0000“QD：FONT.OBJ”

でセーブしてください。

●SUB.OBJ（リスト4）

これは図1にあるサブルーチンパックです。これもBASICのモニタで入力し，

＊SC600 C79A 00 00“QD：SUB.OBJ”

でセーブします。

いよいよ，RUN“ブヒン バコ V2.1A”です。

使い方

まず CTRL キーと ←→ キーでアイコンを選び，F1 キーで処理を開始します。

“QD”　QDのディレクトリを表示し，セーブかロードかを聞いてくるので F1, F2 キーで答えてください。マップ名を入力するとセーブまたはロードできます。

“消しゴム”　カーソルキーで，消したい範囲の左上にカーソルを動かし，Z キーを押し，次に右下にカーソルをもっていき X キーを押すと，消してもいいか聞いてくるので，YかNで答えてください。

コピー　“消しゴム”と同じように，コピーした範囲を指定し，次に移動後の左上にカーソルを動かし，CR キーを押すとコピーができます。

“ペン”　このペンは多機能ですよー。

何を書きたいかを数字キーで答えます。

1）パーツ

パーツを書きたい場所にカーソルを動かし，次に CTRL キーを押しながらカーソルキーでパーツを選びスペースを押すと書き込みます。

2）ライン

次に線のタイプを聞いてきます。

①ごく普通の線
②交点がつながっていない線“┴┐”
③交点がつながっている線“┼”
④バス用の線

図2

図3

図4

図5

図6

F1キーでペンON，F2キーでペンOFF。ペンをONにしてカーソルを動かすと線を引きます。

3）箱

角の形と点線か実線かを聞いてきます。"消しゴム"と同様に箱の大きさを入力して，線の種類を数字で入力してください。CRキーを押すと戻ります。

"キー"　字を書くときに使います。カナも使えます。SHIFTキーで英小文字も使えます。間違えたらカーソルキーで戻し，スペースで消してください。F1で"Ω"F2で"μ"が書けます。CRキーで戻ります。

MAP

"マップ"　全体のレイアウトを見るときに使います。

"プリンタ"　プリントアウトをします。

以上が各アイコンの働きと使い方です。SHIFT＋カーソルキーでマップのスクロールが早送りになります。

なお数字で選ぶところ，Y or Nで答えるところでBREAKキーを押すと処理を中断します。

各アイコンで表示される，メッセージに従ってもらえば簡単に使ってもらえると思います。

他目的への転用法

このプログラムを普通に起動してSHIFT＋BREAKでプログラムを止めて，PCGエディタを起動させます（データの名前にはすべてCRキーを押すこと）PCG NO. 1～144と769～1000を変えずに，他のPCGを必要なキャラクタに変えてください。ただしここでは青（カラーコード1）1色でキャラクタを作ってください。キャラクタの大きさによってPCG NOがちがうので注意が必要です。

キャラクタを作り終わったら，QDにセーブしてから，F10キーでエディタから抜けます（PCGエディタの使い方は，ユーティリティアプリケーション・マニュアルをよく読むこと）。

NEW CR

LIMIT $AFFF CR

を実行して，部品箱のQDを入れ，

LOAD "FONT. OBJ" CR

先ほどのQDを入れ，

LOAD "FONT 1" CR

BYE CR

としてモニタに移り，

＊X D000 EF3F B012 CR

＊S B000 CF51 0000"QD：FONT. OBJ" CR

と入力し，QDにセーブしてください。このプログラムを，もとのプログラムとかえてやればOKです。あとは，それぞれ工夫してみてください。

さあ，この部品箱で，ハードに強くなろう。

リスト1　ブヒン　バコ V2.1A

```
10 ONERRORGOTO "ERROR"
20 LIMIT MAX:IF 24000>=SIZE THEN PRINT"██RUN"+CHR$($22)+"ブヒン バコ V2.1A":PRINT"
CRキ- ヲ オシテクダサイ█":NEWON2
30 PAL,7 :INIT"CRT:I":CLS3:CURSOR7,6:PRINT"CIRCUIT EDITOR":CURSOR14,8:PRINT"THE"
40 FPRINT[18,8],929,930,0,931,932,0,933,934,:FPRINT[18,9],945,946,0,947,948,0,94
9,950,:FPRINT[20,11],951:FOR I=952TO960:FPRINTI:NEXT:CURSOR21,10:PRINT"V2.1A"
50 PRINTTAB(10);"██WAIT A MOMENT!"
60 LIMIT$AFFF:LOAD"FONT.OBJ":USR($B000):LIMIT $C5FF:LOAD"SUB.OBJ"
70 DIM CB(4):DEFAULT"QD:
80 X=1:Y=1:CC=$5A:CB(0)=0:CB(1)=8:CB(2)=$48:CB(3)=$88:CB(4)=$C8
90 FOR I=$C7C5 TO$CFFF:POKE I,0:NEXT
100 PAL,3,0:FOR I=0TO3:FOR O=0TO15:P=(I+20)*40+O+24:POKE $C7C5+P,I*16+O+160:POKE
 $CB85+P,CB(4):NEXTO,I
110 DEF KEY(1)="日":DEF KEY(2)="月":USR($C618)
120 CHAIN "MAIN.BTX"
130 LABEL"ERROR"
140 IF ERL=60 OR ERL=120PRINT"プログラム ノ QD ヲ イレテクレ!":WAIT2000
150 PRINT"██████████HIT ANY KEY"
160 GET K$:IF K$="" THEN 160
170 RESUME
```

リスト2 MAIN.BTX

```
10 ON ERROR GOTO "ERROR"
20 CLS3:POKE$CFF4,CB(0),1,1:USR($C60C)
30 OX=X:OY=Y:GOSUB"XYI"
40 IF I$="■" GOSUB"ALLCLS":CLS
50 IF PEEK($E001)<>191 THEN 80 ELSE X=OX:Y=OY
60 IF I$="■" AND FC<6 THEN FC=FC+1
70 IF I$="■" AND FC>0 THEN FC=FC-1
80 IF I$="■" THEN CURSOR26,0:PRINTMID$("QD      ERASER MOVE   KEY    PEN    MAP
  PRINTER",FC*7+1,7):BEEP:ON FC+1 GOSUB "QD","ELES","MOVE","KEY","WRITE","MAP","
PRINTER":PAL,3,0,3:CLS:BEEP:GOSUB220
90 FOR I=0TO13:IF FC=INT(I/2)THEN CR=7ELSE CR=0
100 CCOLOR 24+I,23,,CR:CCOLOR 24+I,24,,CR:NEXT:GOTO30
110 LABEL "XYI"
120 CURSOR0,0:PRINT"POINT "USING"X= ###";X;:PRINTUSING"  Y= ##";Y
130 IF PEEK($E001)=127 THEN RETURN
140 USR($C60F):I$=CHR$(PEEK($CFF4))
150 IF PEEK($CFF4)=0 THEN 130 ELSE POKE $CFFF+WX+WY*40,0 '
160 IF PEEK($E001)=191 THEN 310 ELSE CX=12:CY=12
170 IF PEEK($E001)=254 THEN CS=5 ELSE CS=1
180 IF I$="■" THEN Y=Y-CS
190 IF I$="■" THEN Y=Y+CS
200 IF I$="■" THEN X=X-CS
210 IF I$="■" THEN X=X+CS
220 IF X>100THEN X=100
230 IF X<1  THEN X=1
240 IF Y>60 THEN Y=60
250 IF Y<1  THEN Y=60
260 SX=X-11:SY=Y-11
270 IF X<12 THEN CX=X   :SX=1
280 IF X>88 THEN CX=X-76:SX=77
290 IF Y<12 THEN CY=Y   :SY=1
300 IF Y>48 THEN CY=Y-36:SY=37
310 POKE $CFFF+CX+CY*40,$EF:WX=CX:WY=CY
320 POKE $CFF5,SX,SY
330 USR($C60C)
340 RETURN
350 LABEL "2XYI"
360 CURSOR24,17:PRINT"START SET (Z)   "
370 GOSUB"XYI":IF I$<>"Z" THEN 370 ELSE X1=X:Y1=Y:POKE$CFF0,X,Y:USR($C603):H1=PE
EK($CFF2):H2=PEEK($CFF3):POKE $CFF2,224,$C8:USR($C600):BEEP
380 CURSOR24,17:PRINT"END   SET (X)   "
390 GOSUB"XYI":IF I$<>"X"THEN 390 ELSE IF X<X1 AND Y<Y1 THEN 390
400 X2=X:Y2=Y:POKE$CFF0,X,Y:USR($C603):H3=PEEK($CFF2):H4=PEEK($CFF3):POKE $CFF2,
225,$C8:USR($C600):BEEP
410 CURSOR24,17:PRINT[7,1]"OK? Y)ES OR N)O":BEEP
420 IF PEEK($E001)=127 THEN I$="Y":GOTO440 ELSE GOSUB"XYI":IF PEEK($CFF4)=0 THEN
 420
430 IF I$<>"N" AND I$<>"Y" THEN 420
440 POKE $CFF0,X1,Y1,H1,H2:USR($C600):POKE $CFF0,X2,Y2,H3,H4:USR($C600)
450 IF I$="Y" THEN:RETURN
460 GOTO360
470 LABEL "BOXW"
480 GOSUB"2XYI":IF PEEK($E001)=127 THEN RETURN ELSE CLS:GOSUB"LTYPE":IF PEEK($E0
01)=127 THEN RETURN
490 Y2=Y2-Y1:X2=X2-X1:FOR I=1TOX2:POKE $CFF0,X1+I,Y1,C1,8:USR($C600):POKE $CFF0,
X1+I,Y1+Y2,C2,8:USR($C600):NEXT
500 FOR I=1TOY2:POKE $CFF0,X1,Y1+I,C3,8:USR($C600):POKE $CFF0,X1+X2,Y1+I,C4,8:US
R($C600):NEXT
510 POKE$CFF0,X1,Y1,C5,8:USR($C600):POKE$CFF0,X1,Y1+Y2,C6,8:USR($C600):POKE$CFF0
,X1+X2,Y1,C7,8:USR($C600):POKE$CFF0,X1+X2,Y1+Y2,C8,8:USR($C600):GOTO"WRITE"
520 LABEL "ELES"
530 GOSUB"2XYI":IF PEEK($E001)=127 THEN RETURN
540 FORXI=X1TOX2:FORYI=Y1TOY2:POKE $CFF0,XI,YI,0,0:USR($C600)
550 CURSOR0,0:PRINT"ERASING "USING"X= ###";XI;:PRINTUSING"  Y= ##";YI
560 NEXT YI,XI
570 RETURN
580 LABEL "MOVE"
590 GOSUB"2XYI":IF PEEK($E001)=127 THEN RETURN ELSE CURSOR26,17:PRINT"SET PONIT(
CR)"
600 GOSUB"XYI":IFI$<>CHR$(13)THEN 600
610 X3=X:Y3=Y
620 X3=X3-X1:Y3=Y3-Y1:X2=X2:Y2=Y2
630 IF0<X3THENXX=X1:X1=X2:X2=XX
640 IF0<Y3THENYY=Y1:Y1=Y2:Y2=YY
650 IFX1>X2THENFOR XI=X1TOX2STEP-1ELSEFORXI=X1TOX2
660 IFY1>Y2THENFOR YI=Y1TOY2STEP-1ELSEFORYI=Y1TOY2
670 IF YI<1 OR YI>60 OR XI<1 OR XI>100THEN 700
680 POKE$CFF0,XI,YI:USR($C603):POKE $CFF0,XI+X3,YI+Y3:USR($C600)
690 CURSOR0,0:PRINT"MOVEING "USING"X= ###";XI;:PRINTUSING"  Y= ##";YI
700 NEXTYI,XI
710 RETURN
720 LABEL "LTYPE"
730 CURSOR24,17:PRINT"CORNER TYPE":PRINTTAB(24);"1)┌┐  2)┌─┐ 3)╭╮":PRINTTAB(26);"
└┘   └─┘   ╰╯"
740 GET I$
750 IF I$="1"THENC1=$18:C2=$18:C3=$19:C4=$19:C5=$54:C6=$64:C7=$55:C8=$65:GOTO800
760 IF I$="2"THENC1=$1C:C2=$1D:C3=$1E:C4=$1F:C5=$58:C6=$68:C7=$59:C8=$69:GOTO800
770 IF I$="3"THENC1=$18:C2=$18:C3=$19:C4=$19:C5=$50:C6=$60:C7=$51:C8=$61:RETURN
780 IF PEEK($E001)=127 THEN RETURN
790 GOTO740
800 CURSOR24,17:PRINT"1) POINT LINE   ";TAB(24);"2) NOMAL LINE   ";TAB(24);"  LI
NE TYPE    "
810 GET I$
820 IF VAL(I$)<4 AND VAL(I$)>0 THEN GOSUB"COLOR"
830 IF I$="1"THEN C1=C1+16:C2=C2+16:C3=C3+16:C4=C4+16:C5=C5-2:C6=C6-2:C7=C7-2:C8
=C8-2:RETURN
840 IF I$="2" OR PEEK($E001)=127 THEN RETURN
```


▶とうとう今年もあと１ヵ月になってしまいました。私は就職も決まり，愛機MZ-2000の前でカチカチやっている今日この頃であります。近頃かぜが流行しているようなので，皆さんも気をつけてください。もしひいてしまったら，必ずネピアで鼻をかんでください。私は王子製紙に入社したのです。　　福地　敏男（17）北海道

```
850 GOTO810
860 LABEL "LINE":LC=1
870 CURSOR24,17:PRINT"1)NOMAL  LINE   ";TAB(24);"2)JUMP   CROSS   ";TAB(24);"3)C
ONTACT CROSS ";TAB(24);"4)BUNCH LINE"
880 GET I$
890 IF VAL(I$)<5 AND VAL(I$)>0 THEN GOSUB"COLOR"
900 IF I$="1" THEN CC=$70:CE=0:CD=0
910 IF I$="2" THEN CC=$11:CE=0:CD=0
920 IF I$="3" THEN CC=$10:CE=2:CD=0
930 IF I$="4" THEN CC=$70:CE=-2:CD=-4
940 IF PEEK($E001)=127 THEN BEEP:GOTO"WRITE"
950 IF VAL(I$)<1 OR VAL(I$)>4 THEN 880
960 CURSOR26,0:PRINT"PEN OFF"
970 GOSUB"XYI":GOSUB220
980 IF I$="月" THEN LC=1:BEEP:CR=0:CURSOR26,0:PRINT"PEN OFF"
990 IF I$="日" THEN LC=0:BEEP:CR=0:CURSOR26,0:PRINT"PEN ON "
1000 IF I$=CHR$(13)THEN CLS:GOTO"WRITE"
1010 IF LC THEN 970
1020 XO=X:YO=Y:IO$=I$
1030 GOSUB"XYI":IF PEEK($CFF4)=0THEN1030
1040 POKE $CFFF+CX+CY*40,$1B
1050 POKE $CFF0,X,Y:USR($C603):CM=PEEK($CFF2)
1060 IF IO$=I$ AND(I$="↑" OR I$="↓")THENCR=$19
1070 IF IO$=I$ AND(I$="←" OR I$="→")THENCR=$18
1080 IFCM=$18 AND CR=$19THENCR=CC:GOTO1230
1090 IFCM=$19 AND CR=$18THENCR=CC:GOTO1230
1100 IF PEEK($E001)<>191 THEN 1150
1110 IFI$="↑"THENCR=$6D-CE:GOTO1240
1120 IFI$="↓"THENCR=$5C-CE:GOTO1240
1130 IFI$="←"THENCR=$5D-CE:GOTO1240
1140 IFI$="→"THENCR=$6C-CE:GOTO1240
1150 IFIO$="→"ANDI$="↑"THENCR=$65+CD:GOTO1240
1160 IFIO$="←"ANDI$="↑"THENCR=$64+CD:GOTO1240
1170 IFIO$="→"ANDI$="↓"THENCR=$55+CD:GOTO1240
1180 IFIO$="←"ANDI$="↓"THENCR=$54+CD:GOTO1240
1190 IFIO$="↓"ANDI$="←"THENCR=$65+CD:GOTO1240
1200 IFIO$="↑"ANDI$="←"THENCR=$55+CD:GOTO1240
1210 IFIO$="↓"ANDI$="→"THENCR=$64+CD:GOTO1240
1220 IFIO$="↑"ANDI$="→"THENCR=$54+CD:GOTO1240
1230 POKE $CFF0,X,Y,CR,8:USR($C600):IO$=I$:GOTO980
1240 POKE $CFF0,XO,YO,CR,8:X=XO:Y=YO:USR($C600):GOTO980
1250 POKE $CFF0,XO,YO,CR,8:USR($C600):GOTO980
1260 LABEL "WRITE"
1270 CURSOR24,17:PRINTTAB(24);"1)PARTS           ";TAB(24);"2)LINE            ";TAB(
24);"3)BOX            ";TAB(24);"4)WRITE END"
1280 GET I$
1290 IF VAL(I$)<5 AND VAL(I$)>0 THEN GOSUB"COLOR"
1300 IF I$="2" THEN "LINE"
1310 IF I$="3" THEN "BOXW"
1320 IF I$="4" RETURN
1330 IF PEEK($E001)=127 THEN RETURN
1340 IF I$<>"1" THEN 1280
1350 CA=0:ZY=1:ZX=1:MY=0:MX=0:POKE$CFF3,$18:USR($C615)
1360 CONSOLE1,16,24,16:XO=X:YO=Y:QX=MX:QY=MY
1370 USR($C60F):IF PEEK($CFF4)<>0THENPOKE$CFFF+WX+WY*40,0,0:POKE$CFFF+WX+(WY+1)*
40,0,0:GOSUB"XYI":CURSOR1,24:PRINT"■" ELSE 1370
1380 IF PEEK($E001)<>191THEN1470ELSE X=XO:Y=YO
1390 IF I$="↑" THEN MY=MY-ZY
1400 IF MY<0 THEN MY=0:CA=ABS((CA-1)MOD3):MX=0
1410 IF I$="↓" THEN MY=MY+ZY
1420 IF MY>15 THEN MY=0:CA=ABS((CA+1)MOD3):MX=0
1430 IF I$="←" AND MX>0 THEN MX=MX-ZX
1440 IF I$="→" AND MX<15THEN MX=MX+ZX
1450 IF CA=2 AND(MY=5 OR MY=6)THEN MX=0
1460 POKE $CFF3,CB(CA+1):USR($C615)
1470 IF CA=0 THEN ZX=1:ZY=1:POKE $D018+MX+(MY+1)*40,$EF
1480 IF CA=1 THEN POKE $CFFF+CX+CY*40,$EF,$EF:POKE $CFFF+CX+(CY+1)*40,$EF,$EF:ZX
=2:ZY=2:POKE $D018+MX+(MY+1)*40,$EF,$EF:POKE$D018+MX+(MY+2)*40,$EF,$EF
1490 IF CA=2 AND MY<=6 THEN POKE $CFFF+CX+CY*40,$EF,$EF:ZX=2:ZY=1:POKE $D018+MX+
(MY+1)*40,$EF,$EF
1500 IF CA=2 AND MY>6 THEN POKE$CFFF+CX+(CY+1)*40,$EF:ZX=1:ZY=2:POKE$D018+MX+(MY
+1)*40,$EF:POKE$D018+MX+(MY+2)*40,$EF
1510 IF I$=CHR$(13)THEN CONSOLE:GOTO"WRITE"
1520 IF I$<>" " THEN 1360ELSE BEEP
1530 CR=MX+MY*16
1540 IF CA=0 THEN POKE $CFF0,X,Y,CR,8:USR($C600)
1550 IF CA=1 THEN POKE $CFF0,X,Y,CR,$48:USR($C600):POKE $CFF0,X+1,Y,CR+1,$48:USR
($C600):POKE $CFF0,X,Y+1,CR+16,$48:USR($C600):POKE $CFF0,X+1,Y+1,CR+17,$48:USR($
C600)
1560 IF CA=2 AND MY<=7 THEN POKE $CFF0,X,Y,CR,$88:USR($C600):POKE $CFF0,X+1,Y,CR
+1,$88:USR($C600)
1570 IF CA=2 AND MY>=6 THEN POKE $CFF0,X,Y,CR,$88:USR($C600):POKE $CFF0,X,Y+1,CR
+16,$88:USR($C600)
1580 GOTO1360
1590 LABEL "KEY"
1600 CURSOR26,17:PRINT"ALP"
1610 GOSUB"XYI"
1620 IF PEEK($CFF4)=0 THEN 1610
1630 IF I$=CHR$(13):PRINTCHR$(25):RETURN
1640 IFX>100THENX=0ELSECH=-1
1650 CH=-1
1660 IF ASC(I$)<=24THEN 1610
1670 IF I$=CHR$(26) THEN PRINTCHR$(26):BEEP:CURSOR26,17:PRINT"カ ナ":GOTO1610
1680 IF I$=CHR$(25) THEN PRINTCHR$(25):BEEP:CURSOR26,17:PRINT"ALP":GOTO1610
1690 IFI$>="A" ANDI$<="Z"THENCH=ASC(I$)-65:IF PEEK($E001)=254THENCH=CH+26
1700 IFI$>="0" ANDI$<="9"THENCH=ASC(I$)+4
1710 IFI$>="ｦ" ANDI$<="ﾟ"THENCH=ASC(I$)-72
1720 IFI$="-"THEN CH=128
1730 IFI$="+"THEN CH=129
1740 IFI$="/"THEN CH=130
1750 IFI$="*"THEN CH=131
1760 IFI$="日"THEN CH=132
1770 IFI$="月"THEN CH=133
```

▶最近，大好きな女性ができてしまったせいか，ターボ君にさわる機会が非常に少なくなってきている。だから，うちのターボ君は主人の言うことをなかなか聞いてくれない。ホントに焼きもちやきで困ってしまいます。　高津　誠（26）愛知県

```
1780 IFI$="."THEN CH=144
1790 IFI$=","THEN CH=145
1800 IFI$="("THEN CH=146
1810 IFI$=")"THEN CH=147
1820 IFI$="$"THEN CH=148
1830 IFI$="%"THEN CH=149
1840 IFI$="&"THEN CH=150
1850 IFI$="="THEN CH=151
1860 IFI$=" "THEN POKE $CFF0,X,Y,0,0:USR($C600):X=X+1:GOTO1890
1870 IFCH=-1 THEN BEEP:GOTO1610
1880 POKE $CFF0,X,Y,CH,$C8:USR($C600):X=X+1
1890 POKE$CFFF+CX+CY*40,0:GOSUB220:GOTO1610
1900 LABEL "QD"
1910 PAL,0,0,0:PRINT"■■■■■QD MODE"
1920 DIR:PRINTTAB(7);"SAVE OR LOAD?":PRINTTAB(3)"(F1 OR F2 )"
1930 GET K$:IF K$="В"THEN 1970
1940 IFK$="Я"THEN 1990
1950 IF PEEK($E001)=127 THEN RETURN
1960 GOTO 1930
1970 PRINT"  SAVE!":INPUT"MAP NAME?";FL$:IFLEN(FL$)>11 THEN FL$=LEFT$(FL$,16)
1980 USR($C609,FL$):PAL,3:RETURN
1990 PRINT"  LOAD!":INPUT"MAP NAME?";FL$:IFLEN(FL$)>11 THEN FL$=LEFT$(FL$,16)
2000 USR($C606,FL$):PAL,3:RETURN
2010 LABEL "MAP"
2020 PAL,0,7,7:CLS:FPRINT[0,0]:FOR I=1TO1000:FPRINTI:NEXT:PRINT"■■■■■■MAP OF L
AYOUT":POKE $108D,2,1
2030 USR($C612):POKE $CFF3,0
2040 CURSOR 10,23:PRINT"HIT ANY KEY."
2050 GET I$:IF I$="" THEN 2050
2060 PAL,3,0,3:CLS2:USR($C615):RETURN
2070 LABEL "ALLCLS"
2080 CLS3:PRINT"■■ALL CLEAR OF MAP"
2090 PRINTTAB(10);"OK?"
2100 GET I$:IFI$="" THEN2100
2110 IF I$="Y"THEN USR($C618)
2120 USR($C60C):RETURN
2130 LABEL "PRINTER"
2140 PRINT/P CHR$($9)
2150 FOR XI=100TO1STEP-1:PRINT/P CHR$($1B,$18,$E0,$1);:FOR YI=1TO60
2160 CURSOR0,0:PRINT"PRINTING "USING"X= ###";XI;:PRINTUSING"  Y= ##";YI
2170 POKE $CFF0,XI,YI:USR($C603):CN=PEEK($CFF2)
2180 IF PEEK($CFF3)=0   THEN A$=HEXCHR$("0000000000000000"):GOTO2240
2190 IF PEEK($CFF3)= 8  THEN CR=CN+1
2200 IF PEEK($CFF3)=$48 THEN CR=257+CN
2210 IF PEEK($CFF3)=$88 THEN CR=513+CN
2220 IF PEEK($CFF3)=$C8 THEN CR=769+CN
2230 A$=LEFT$(FONT$(2,CR),8)
2240 USR($C61B,A$)
2245 NEXT YI:NEXT XI:RETURN
2250 LABEL "COLOR"
2260 CO=VAL(I$)
2270 FOR CU=1TO4:IF CO=CU THEN CP=7 ELSE CP=2
2280 FOR CI=24TO39:CCOLORCI,16+CU,2,CP:NEXT CI,CU
2290 RETURN
2300 LABEL "ERROR"
2310 FORI=1TO5:WAIT300:PRINT"■■■■■■■■■■■■■CAUTION !!":BEEP:WAIT250:PRINT"■■■■■■■■
■■■■          ■":WAIT300:NEXT:PRINTTAB(5);
2320 IF ERN=46 PRINT"QD WRITE PROTECT"
2330 IF ERN=53 PRINT"TOO MANY MAPS"
2340 IF ERN=40 PRINT"MAP FILE NOT FOUND"
2350 IF ERN=42 PRINT"ILLEGAL MAP'S NAME"
2360 IF ERN=50 PRINT"QD OFFLINE"
2370 IF ERN=60 PRINT"ALREADY EXIST"
2380 IF ERN=61 PRINT"NOT MAP'S FILE"
2390 IF ERN=65 PRINT"PRINTER OFFLINE":WAIT2000:RESUME20
2400 PRINTTAB(10);"HIT ANY KEY"
2410 GET K$:IF K$="" THEN 2410
2420 RESUME "QD"
```

印字サンプル(MZ-1P07)

▶12月STUDIO MZ p.179の沼野さん，私もフル装備の1500を使っていますが，RAMファイルの調子がよくないのはボード上のクリスタル不良です。私も同じことを経験しているので，保障の効いているうちに早く修理に出したほうがいいですよ。

久野　展宏（21）愛知県

リスト3 FONT.OBJ

```
B000 21 12 B0 11 00 D0 01 00 :C5
B008 20 3E 01 D3 E5 ED B0 D3 :87
B010 E6 C9 00 00 00 00 00 00 :AF
B018 00 00 00 00 00 00 00 00 :00
B020 00 00 00 00 00 00 00 00 :00
B028 00 00 00 00 00 00 00 00 :00
B030 00 00 00 00 00 00 00 00 :00
B038 00 00 00 00 00 00 00 00 :00
B040 00 00 00 00 00 00 00 00 :00
B048 00 00 00 00 00 00 00 00 :00
B050 00 00 00 00 00 00 00 00 :00
B058 00 00 00 00 00 00 00 00 :00
B060 00 00 00 00 00 00 00 00 :00
B068 00 00 00 00 00 00 00 00 :00
B070 00 00 00 00 00 00 00 00 :00
B078 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: 27 19 B1 E4 E5 BD B1 D3 :FB

B080 00 00 00 00 00 00 00 00 :00
B088 00 00 00 00 00 00 00 00 :00
B090 00 00 08 08 1C FF 1C 08 :4F
B098 08 08 08 1C 2A C9 08 08 :37
B0A0 08 08 08 08 08 08 3E 1C :8A
B0A8 08 00 00 08 1C 3E 08 08 :7A
B0B0 08 08 3E 1C 08 FF 00 00 :71
B0B8 00 00 08 08 09 0B 0F 0B :3E
B0C0 09 08 22 14 08 FF 00 00 :4E
B0C8 00 00 08 08 09 0A 0C 0A :39
B0D0 09 08 00 00 00 FF 00 00 :10
B0D8 00 00 08 08 08 08 08 08 :30
B0E0 08 08 00 04 08 FF 20 40 :7B
B0E8 00 00 08 0A 0C 08 18 28 :66
B0F0 08 08 FF 00 00 00 00 00 :0F
B0F8 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: 42 38 97 8A A8 2F C5 B9 :F0

B100 00 FF 80 80 80 80 80 80 :FF
B108 80 80 01 01 01 01 01 01 :06
B110 01 01 08 14 08 08 08 08 :3E
B118 08 08 00 00 02 FD 02 00 :11
B120 00 00 00 00 40 BF 40 00 :3F
B128 00 00 08 08 08 08 08 08 :30
B130 14 08 00 00 00 FF 08 14 :37
B138 08 08 08 08 28 D8 28 08 :50
B140 08 08 08 14 08 FF 00 00 :33
B148 00 00 08 08 0A 0D 0A 08 :39
B150 08 08 00 00 00 99 00 00 :A9
B158 00 00 08 00 00 08 08 00 :18
B160 00 08 08 08 08 08 7F 55 :FC
B168 2A 54 00 00 00 00 00 00 :7E
B170 00 00 99 00 00 00 00 00 :99
B178 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: DF 04 52 C9 15 D9 94 0A :8A

B180 00 99 80 00 00 80 80 00 :19
B188 00 80 01 00 00 01 01 00 :83
B190 00 01 C0 30 0C 03 00 00 :00
B198 00 00 00 00 00 00 00 00 :00
B1A0 00 00 03 0C 30 C0 00 00 :FF
B1A8 00 00 00 00 00 00 00 00 :00
B1B0 00 00 00 00 00 03 0C 10 :1F
B1B8 20 C0 00 00 00 00 00 00 :E0
B1C0 00 00 00 00 00 C0 30 08 :F8
B1C8 04 03 00 00 00 00 00 00 :07
B1D0 00 00 08 08 10 20 40 40 :C0
B1D8 80 80 08 08 04 04 02 02 :1C
B1E0 01 01 00 00 00 00 00 00 :02
B1E8 00 00 01 01 02 02 04 04 :0E
B1F0 08 08 00 00 00 00 00 00 :10
B1F8 00 00 80 80 40 20 10 10 :80
--------------------------------
SUM: AD 66 D5 CD 92 4D 13 6E :15

B200 08 08 00 00 00 00 00 00 :10
B208 00 00 00 00 00 00 00 00 :00
B210 00 00 00 00 00 00 00 00 :00
B218 00 00 00 00 00 00 00 00 :00
B220 00 00 00 00 00 00 00 00 :00
B228 00 00 00 00 00 00 00 00 :00
B230 00 00 00 00 00 00 00 00 :00
B238 00 00 00 00 00 00 00 00 :00
B240 00 00 00 00 00 00 00 00 :00
B248 00 00 00 00 00 00 00 00 :00
B250 00 00 00 00 00 00 00 00 :00
B258 00 00 00 00 00 00 00 00 :00
B260 00 00 00 00 00 00 00 00 :00
B268 00 00 00 00 00 00 00 00 :00
B270 00 00 00 00 00 00 00 00 :00
B278 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: 08 08 00 00 00 00 00 00 :10

B280 00 00 00 00 00 00 00 00 :00
B288 00 00 00 00 00 00 00 00 :00
B290 00 00 00 00 00 03 04 08 :0F
B298 08 08 00 00 00 E0 10 08 :08
B2A0 08 08 00 00 00 09 08 00 :21
B2A8 00 08 00 00 00 98 08 00 :A8
B2B0 00 08 00 00 00 0F 08 08 :27
B2B8 08 08 00 00 00 F8 08 08 :18
B2C0 08 08 99 00 80 80 00 00 :A9
B2C8 80 80 99 00 00 01 01 00 :9B
B2D0 00 01 FF 80 80 80 80 80 :80
B2D8 80 80 FF 01 01 01 01 01 :04
B2E0 01 01 00 00 1C FF 1C 08 :41
B2E8 08 08 08 08 1C FC 1C 08 :5C
B2F0 08 08 00 00 00 FF 08 08 :1F
B2F8 08 08 08 08 08 F8 08 08 :30
--------------------------------
SUM: 39 4A 40 91 41 7F FE C1 :D3

B300 08 08 00 00 00 FF 04 08 :1B
B308 08 08 08 08 08 E8 18 08 :30
B310 08 08 08 08 04 03 00 00 :27
B318 00 00 08 08 10 E0 00 00 :00
B320 00 00 08 00 00 09 00 00 :11
B328 00 00 08 00 00 98 00 00 :A0
B330 00 00 08 08 08 0F 00 00 :27
B338 00 00 08 08 08 F8 00 00 :10
B340 00 00 00 00 80 80 00 00 :00
B348 80 99 01 00 00 01 01 00 :1C
B350 00 99 80 80 80 80 80 80 :99
B358 80 FF 01 01 01 01 01 01 :85
B360 01 FF 08 08 1C 1F 1C 08 :6F
B368 08 08 08 08 1C FF 1C 00 :57
B370 00 00 08 08 08 0F 08 08 :37
B378 08 08 08 08 08 FF 00 00 :27
--------------------------------
SUM: 29 58 DA C9 75 A0 DE A1 :B8

B380 00 00 08 08 0C 0B 08 08 :37
B388 08 08 08 08 10 FF 00 00 :2F
B390 00 00 08 08 08 FF 08 08 :27
B398 08 08 00 00 18 24 24 18 :88
B3A0 00 00 00 00 00 00 00 00 :00
B3A8 00 00 00 00 00 00 00 00 :00
B3B0 00 00 00 00 00 00 00 00 :00
B3B8 00 00 00 00 00 00 00 00 :00
B3C0 00 00 00 00 00 00 00 00 :00
B3C8 00 00 00 00 00 00 00 00 :00
B3D0 00 00 00 00 00 00 00 00 :00
B3D8 00 00 00 00 00 00 00 00 :00
B3E0 00 00 00 00 00 00 00 00 :00
B3E8 00 00 00 00 00 00 00 00 :00
B3F0 00 00 08 0C 0A 0B 0C 08 :3D
B3F8 08 08 00 00 00 FF 12 0C :2D
--------------------------------
SUM: 18 18 20 24 46 37 52 3C :7F

B400 08 08 00 00 00 FF 10 08 :27
B408 08 08 08 08 18 E8 08 08 :30
B410 08 08 00 00 00 00 00 00 :10
B418 00 00 00 00 00 00 00 00 :00
B420 00 00 00 00 00 00 00 00 :00
B428 00 00 00 00 00 00 00 00 :00
B430 00 00 00 00 00 00 00 00 :00
B438 00 00 00 00 00 00 00 00 :00
B440 00 00 00 00 00 00 00 00 :00
B448 00 00 00 00 00 00 00 00 :00
B450 00 00 00 00 00 00 00 00 :00
B458 00 00 00 00 00 00 00 00 :00
B460 00 00 00 00 00 00 00 00 :00
B468 00 00 00 00 00 00 00 00 :00
B470 00 00 08 18 24 FF 00 00 :43
B478 00 00 08 08 18 E8 28 18 :50
--------------------------------
SUM: 18 18 18 28 54 CE 40 28 :FA

B480 08 08 08 08 08 0B 0C 08 :47
B488 08 08 08 08 04 FF 00 00 :23
B490 00 00 08 14 08 14 22 41 :9B
B498 7F 08 60 50 4A C5 4A 50 :E0
B4A0 60 00 08 7F 41 22 14 08 :66
B4A8 14 08 06 0A 52 A3 52 0A :7D
B4B0 06 00 00 00 00 00 00 00 :06
B4B8 00 00 08 08 08 14 22 41 :8F
B4C0 7F 08 60 50 48 C7 48 50 :DE
B4C8 60 00 08 7F 41 22 14 08 :66
B4D0 08 08 06 0A 12 E3 12 0A :31
B4D8 06 00 00 00 00 00 00 00 :06
B4E0 00 00 00 00 00 00 00 00 :00
B4E8 00 00 00 00 00 00 00 00 :00
B4F0 00 00 00 00 00 00 00 00 :00
B4F8 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: F6 30 FC DE 94 88 6E 4E :D8
```

(B500～B57Fは00で埋めてください)

```
B580 00 00 00 00 00 00 00 00 :00
B588 00 00 00 00 00 00 00 00 :00
B590 00 00 08 08 7F 08 14 22 :CD
B598 41 7F C4 A4 94 8F 94 A4 :83
B5A0 C4 00 7F 41 22 14 08 7F :41
B5A8 08 08 23 25 29 F1 29 25 :C0
B5B0 23 00 00 00 00 00 00 00 :23
B5B8 00 00 08 48 7F 09 14 22 :0E
B5C0 41 7F C6 A4 94 8F 94 A4 :85
B5C8 CC 00 7F 41 22 14 09 7F :4A
B5D0 48 08 63 25 29 F1 29 25 :40
B5D8 33 00 00 00 00 00 00 00 :33
B5E0 00 00 00 24 24 E7 24 24 :77
B5E8 00 00 08 08 3E 00 00 3E :8C
B5F0 08 08 00 00 00 00 00 00 :10
B5F8 00 00 00 34 2C F7 2C 34 :B7
--------------------------------
SUM: C0 16 26 C4 4A 17 03 6A :8E

B600 00 00 08 08 3E 14 2A 3E :CA
B608 08 08 00 00 00 00 00 00 :10
B610 00 00 00 00 00 00 00 00 :00
B618 00 00 00 00 00 00 00 00 :00
B620 00 00 00 00 00 00 00 00 :00
B628 00 00 00 00 00 00 00 00 :00
B630 00 00 00 00 00 00 00 00 :00
B638 00 00 00 00 00 00 00 00 :00
B640 00 00 00 00 00 00 00 00 :00
B648 00 00 00 00 00 00 00 00 :00
B650 00 00 00 00 00 00 00 00 :00
B658 00 00 00 00 00 00 00 00 :00
B660 00 00 00 00 00 00 00 00 :00
B668 00 00 00 00 00 00 00 00 :00
B670 00 00 00 00 00 00 00 00 :00
B678 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: 08 08 08 08 3E 14 2A 3E :DA

B680 00 00 00 00 00 00 00 00 :00
B688 00 00 00 00 00 00 00 00 :00
B690 00 00 08 08 0B 3C 00 00 :54
B698 00 FF 00 00 00 3C 00 00 :3B
B6A0 00 FF 00 00 00 00 00 00 :FF
B6A8 00 00 80 80 88 8F 88 88 :27
B6B0 80 80 80 80 88 88 88 88 :20
B6B8 80 80 00 00 00 00 00 00 :00
B6C0 00 00 18 FF 00 42 A5 42 :40
B6C8 00 00 00 00 00 00 00 00 :00
B6D0 00 00 40 30 0B 46 A5 42 :A5
B6D8 00 00 00 00 00 00 00 00 :00
B6E0 00 00 48 54 48 C0 C0 48 :AC
B6E8 54 48 00 00 00 00 00 00 :9C
B6F0 00 00 08 14 18 20 40 48 :DC
B6F8 94 08 00 00 00 00 00 00 :9C
--------------------------------
SUM: E8 4E B0 9F 80 F7 5A 24 :7A
```

(B700～B7FFは00で埋めてください)

```
B800 00 00 00 00 00 00 00 00 :00
B808 00 00 00 00 00 00 00 00 :00
B810 00 00 07 19 21 41 41 81 :44
B818 9F 83 C0 30 08 04 04 02 :24
B820 F2 82 07 18 20 FC 42 81 :72
B828 80 87 E0 18 04 42 42 41 :C8
B830 C1 FF 0B 0C 18 28 28 48 :87
B838 44 42 E8 18 0C 0A 0A 89 :2F
B840 91 A1 07 18 20 43 43 83 :7A
B848 83 FF E0 18 04 3F 42 81 :80
B850 E1 01 07 18 20 40 40 8F :30
B858 82 84 E0 98 84 82 82 F9 :FF
B860 41 29 03 0C 10 FF 20 40 :E8
B868 40 40 E0 18 04 22 A2 61 :A1
B870 21 3F 10 17 18 30 50 50 :6F
B878 9C 98 10 F0 18 14 12 12 :84
--------------------------------
SUM: CB 32 72 A8 7D 5E 66 A5 :FD

B880 11 11 07 18 20 44 44 85 :6E
B888 86 FC C0 30 08 7F 84 02 :7F
B890 02 02 85 89 91 50 50 30 :73
```

▶毎月，Oh!MZの発売日の18日が待ちどおしくて，11月のように17日が日曜日だったりすると，「どっかの本屋さんが一日早いけど，うっかりして店頭に並べているかもしれない」と考え，つい本屋さんめぐりをしてしまうのでした。　山本　勇（36）千葉県

```
B898 18 17 42 22 12 14 14 18 :E5
B8A0 30 D0 80 81 42 FC 20 18 :77
B8A8 07 00 C1 41 42 42 04 18 :A9
B8B0 E0 00 41 4F 40 20 20 10 :00
B8B8 0C 03 C1 F9 81 82 82 84 :D2
B8C0 98 E0 83 82 42 42 20 18 :39
B8C8 07 00 01 81 42 3F 04 18 :26
B8D0 E0 00 88 88 48 48 28 18 :C0
B8D8 0F 08 19 39 0A 0A 0C 18 :A1
B8E0 E8 08 40 42 23 FF 10 0C :B0
B8E8 03 00 61 A1 22 A2 04 18 :E5
B8F0 E0 00 94 82 9F 41 41 21 :3B
B8F8 19 07 21 41 F1 02 02 04 :7B
---------------------------------
SUM: 46 F0 4C 67 BB BE A1 3C :3F

B900 18 E0 86 85 44 45 20 18 :C4
B908 07 00 02 42 C4 FF 08 30 :46
B910 C0 00 01 01 02 01 07 08 :D4
B918 14 13 00 00 80 80 E0 10 :17
B920 28 C8 00 00 FF 80 40 20 :CF
B928 20 10 00 00 00 C0 20 50 :60
B930 90 8C 30 28 26 21 20 20 :FB
B938 20 20 0C 14 64 84 04 04 :50
B940 04 04 00 00 00 03 04 0A :19
B948 09 39 00 00 FF 01 02 04 :48
B950 04 08 01 01 01 01 07 08 :1F
B958 14 13 00 00 00 80 E0 10 :97
B960 28 C8 00 00 FF 80 40 20 :CF
B968 20 10 00 00 00 C0 20 50 :60
B970 90 9F 30 28 26 21 20 20 :0E
B978 20 20 0C 14 64 84 04 04 :50
---------------------------------
SUM: 08 66 02 41 9C 14 04 AE :13

B980 04 04 00 00 00 03 04 0A :19
B988 09 19 00 00 FF 01 02 04 :28
B990 04 08 20 20 20 20 21 26 :D3
B998 28 30 04 04 04 04 84 64 :50
B9A0 14 0C 10 20 20 40 80 FF :2F
B9A8 00 00 8B 94 50 20 C0 00 :4F
B9B0 00 00 13 14 08 07 01 02 :39
B9B8 01 01 C8 28 10 E0 80 80 :E2
B9C0 00 00 D9 29 0A 04 03 00 :13
B9C8 00 00 08 04 04 02 01 FF :12
B9D0 00 00 20 20 20 20 21 26 :C7
B9D8 28 30 04 04 04 04 84 64 :50
B9E0 14 0C 10 20 20 40 80 FF :2F
B9E8 00 00 98 90 50 20 C0 00 :58
B9F0 00 00 13 14 08 07 01 00 :37
B9F8 00 00 C8 28 10 E0 80 80 :E0
---------------------------------
SUM: 8A 9E 22 51 65 E0 D6 21 :D7

BA00 80 80 F9 09 0A 04 03 00 :13
BA08 00 00 08 04 04 02 01 FF :12
BA10 00 00 01 01 02 01 06 08 :13
BA18 10 10 00 00 80 80 60 10 :90
BA20 08 08 00 00 FF 80 80 80 :8F
BA28 80 80 00 00 00 C0 20 10 :F0
BA30 10 0C 3F 20 20 20 20 20 :FB
BA38 20 20 FC 04 04 04 04 04 :50
BA40 04 04 00 00 00 03 04 08 :17
BA48 08 38 00 00 FF 01 01 01 :42
BA50 01 01 01 01 02 01 07 08 :16
BA58 10 10 00 00 80 80 E0 10 :10
BA60 18 78 00 00 FF 80 FC 92 :9D
BA68 92 89 00 00 00 C0 20 10 :0B
BA70 10 1C 3F 20 20 20 21 2E :1A
BA78 28 29 FC 14 14 74 94 14 :91
---------------------------------
SUM: 47 D7 79 67 67 44 EB D0 :64

BA80 64 84 00 00 00 03 04 08 :F7
BA88 08 38 00 00 FF 01 3F 49 :C8
BA90 49 91 20 20 20 20 20 20 :9A
BA98 20 3F 04 04 04 04 04 04 :77
BAA0 04 FC 80 80 80 80 80 FF :7F
BAA8 00 00 0B 14 10 20 C0 00 :0F
BAB0 00 00 10 10 08 07 01 01 :31
BAB8 00 00 08 08 10 E0 80 40 :C0
BAC0 80 80 DB 28 08 04 03 00 :0F
BAC8 00 00 01 01 01 01 01 FF :04
BAD0 00 00 21 26 28 29 2E 28 :EE
BAD8 28 3F 94 14 74 84 04 04 :0F
BAE0 04 FC B9 84 84 87 80 FF :97
BAE8 00 00 1B 94 90 E0 C0 00 :DF
BAF0 00 00 1E 18 08 07 01 02 :48
BAF8 01 01 08 08 10 E0 80 80 :02
---------------------------------
SUM: 86 44 1F 6B 9C AF 1F 61 :1F

BB00 00 00 D8 29 09 07 03 00 :14
BB08 00 00 91 21 21 E1 01 FF :B4
BB10 00 00 01 01 02 01 07 08 :14
BB18 10 10 00 00 80 80 E0 10 :10
BB20 28 E8 00 00 FF B0 7C 52 :5D
BB28 32 29 00 00 00 C0 20 10 :4B
BB30 10 1C 30 28 27 20 20 23 :0E
BB38 24 24 0C 34 D4 74 94 14 :78
BB40 64 84 00 00 00 03 04 08 :F7
BB48 08 38 00 00 FF 01 3E 4A :C8
BB50 4C 94 01 01 01 01 07 08 :F3
BB58 10 10 00 00 00 80 E0 10 :90
BB60 08 08 00 00 FF 80 80 80 :8F
BB68 80 80 00 00 00 C0 20 10 :F0
BB70 10 1F 3F 20 20 20 20 20 :0E
BB78 20 20 FC 04 04 04 04 04 :50
---------------------------------
SUM: 1E 88 E2 CC C9 26 28 CE :39

BB80 04 04 00 00 00 03 04 08 :17
BB88 08 18 00 00 FF 01 01 01 :22
BB90 01 01 21 26 28 29 2E 2B :F3
BB98 2C 30 24 24 C4 04 04 E4 :54
BBA0 14 0C 24 24 23 40 80 FF :4A
BBA8 00 00 9B 94 D0 20 C0 00 :DF
BBB0 00 00 17 14 08 07 01 02 :3D
BBB8 01 01 08 08 10 E0 80 80 :02
BBC0 00 00 D9 29 0B 04 03 00 :14
BBC8 00 00 24 24 C4 02 01 FF :0E
BBD0 00 00 20 20 20 20 20 20 :C0
BBD8 20 3F 04 04 04 04 04 04 :77
BBE0 04 FC 80 80 80 80 80 FF :7F
BBE8 00 00 18 10 10 20 C0 00 :18
BBF0 00 00 10 10 08 07 01 00 :30
BBF8 00 00 08 08 10 E0 80 80 :00
---------------------------------
SUM: 72 95 F4 37 91 29 E1 3B :08

BC00 80 80 F8 08 08 04 03 00 :0F
BC08 00 00 01 01 01 01 01 FF :04
BC10 00 00 01 01 01 01 07 08 :13
BC18 14 13 00 00 00 80 E0 10 :97
BC20 28 C8 00 00 FF 80 80 80 :6F
BC28 80 80 00 00 00 C0 20 50 :30
BC30 90 9F 3F 20 20 20 20 20 :0E
BC38 20 20 FC 04 04 04 04 04 :50
BC40 04 04 00 00 00 03 04 0A :19
BC48 09 11 00 00 FF 01 01 01 :1C
BC50 01 01 01 01 01 02 02 02 :0A
BC58 04 04 00 00 00 80 40 40 :08
BC60 20 20 00 00 F0 8C A3 F0 :4F
BC68 A0 80 00 00 00 00 00 C0 :E0
BC70 30 0F 3F 20 2E 20 10 10 :0C
BC78 08 08 FC 24 74 24 08 08 :D8
---------------------------------
SUM: F6 6B 71 73 BF 3F B1 20 :14

BC80 10 10 00 00 00 00 00 03 :23
BC88 0C 10 00 00 0F 31 C5 05 :26
BC90 05 01 20 20 20 20 20 20 :C6
BC98 20 3F 04 04 04 04 04 04 :77
BCA0 04 FC 80 80 80 80 80 FF :7F
BCA8 00 00 98 90 50 20 C0 00 :58
BCB0 00 00 13 14 08 07 01 00 :37
BCB8 00 00 C8 28 10 E0 80 80 :E0
BCC0 80 80 F9 09 0A 04 03 00 :13
BCC8 00 00 01 01 01 01 01 FF :04
BCD0 00 00 08 08 10 10 24 2E :B2
BCD8 24 3F 10 10 08 08 04 74 :0B
BCE0 04 FC 80 A0 A0 A3 8C F0 :DF
BCE8 00 00 08 30 C0 00 00 00 :F8
BCF0 00 00 04 04 02 02 01 00 :0D
BCF8 00 00 20 20 40 40 80 80 :C0
---------------------------------
SUM: ED 17 D5 86 E0 DE E3 BC :BC

BD00 80 80 F0 0C 03 00 00 00 :FF
BD08 00 00 01 05 0F C5 31 0F :1A
BD10 00 00 07 18 20 40 40 BF :4E
BD18 82 84 E0 18 04 02 02 F9 :FF
BD20 41 29 03 0C 10 FF 20 40 :EB
BD28 40 40 E0 18 04 22 A2 61 :A1
BD30 21 21 10 17 18 30 50 50 :51
BD38 9C 98 10 F0 18 14 12 12 :84
BD40 11 11 07 18 20 44 44 85 :6E
BD48 86 84 C0 30 08 7F 84 02 :07
BD50 02 02 01 01 01 01 01 01 :0A
BD58 02 02 00 00 00 00 00 80 :84
BD60 40 40 00 00 C0 B0 8C 83 :FF
BD68 82 84 00 00 00 00 00 00 :06
BD70 C0 3F 3F 20 10 10 08 0D :93
BD78 06 04 FC 04 08 08 10 90 :BA
---------------------------------
SUM: 63 C6 DE D9 7B F8 04 C2 :19

BD80 60 20 00 00 00 00 00 00 :80
BD88 03 04 00 00 03 0D 31 E1 :29
BD90 41 21 88 88 48 48 28 18 :42
BD98 0F 08 19 39 0A 0A 0C 18 :A1
BDA0 E8 08 40 42 23 FF 10 0C :B0
BDA8 03 00 61 A1 22 A2 04 18 :E5
BDB0 E0 00 94 82 9F 40 40 20 :35
BDB8 18 07 21 41 F1 02 02 04 :7A
BDC0 18 E0 86 85 44 45 20 18 :C4
BDC8 07 00 02 42 C4 FF 08 30 :46
BDD0 C0 00 04 06 09 08 10 10 :FB
BDD8 20 3F 20 60 B0 10 08 08 :AF
BDE0 04 FC 84 82 87 8C B0 C0 :B9
BDE8 00 00 20 C0 00 00 00 00 :E0
BDF0 00 00 02 02 01 00 00 00 :05
BDF8 00 00 40 40 80 80 80 80 :80
---------------------------------
SUM: 99 77 89 18 F3 AA 2B F9 :72

BE00 80 80 FC 03 00 00 00 00 :FF
BE08 00 00 21 41 C1 31 0D 03 :64
BE10 00 00 01 01 07 08 14 13 :38
BE18 20 20 00 80 E0 10 28 C8 :A0
BE20 04 04 00 00 BF A0 D0 28 :5F
BE28 28 14 00 00 C0 30 08 14 :48
BE30 24 27 3B 08 36 29 26 21 :31
BE38 20 20 1C 10 6C 94 64 84 :54
BE40 04 04 00 00 03 0C 10 28 :4F
BE48 24 64 00 00 FD 05 0B 14 :A9
BE50 14 28 01 01 07 08 10 10 :6D
BE58 20 20 00 80 E0 10 08 08 :C0
BE60 04 04 00 00 BF A0 D0 28 :5F
BE68 28 14 00 00 C0 30 08 04 :38
BE70 04 07 3B 08 36 29 26 21 :F1
BE78 20 20 1C 10 6C 94 64 84 :54
---------------------------------
SUM: BC EE C7 76 D1 8C 40 E4 :68

BE80 04 04 00 00 03 0C 10 20 :47
BE88 20 60 04 04 FD 05 0B 14 :A9
BE90 14 28 20 20 21 26 29 36 :22
BE98 08 38 04 04 84 64 94 6C :30
BEA0 10 1C 14 28 28 D0 A0 BF :BF
BEA8 00 00 26 24 14 08 30 C0 :56
BEB0 00 00 20 20 13 14 08 07 :76
BEB8 01 00 04 04 C8 28 10 E0 :E9
BEC0 80 80 E4 24 28 10 0C 03 :4F
BEC8 00 00 28 14 14 0B 05 FD :5D
BED0 00 00 20 20 21 26 29 36 :E6
BED8 08 38 04 04 84 64 94 6C :30
BEE0 10 1C 14 28 28 D0 A0 BF :BF
BEE8 00 00 06 04 04 08 30 C0 :06
BEF0 00 00 20 20 10 10 08 07 :6F
BEF8 01 00 04 04 08 08 10 E0 :09
---------------------------------
SUM: EA B4 F4 44 E1 44 76 44 :B5

BF00 80 80 E0 20 20 10 0C 03 :3F
BF08 00 00 28 14 14 0B 05 FD :5D
BF10 04 04 01 01 01 02 02 01 :10
BF18 02 02 00 00 00 80 80 80 :84
BF20 40 40 00 00 C0 B0 8C 83 :FF
BF28 F8 A4 00 00 00 00 00 00 :9C
BF30 DB 27 3F 20 10 11 09 09 :91
BF38 05 04 FC 44 C8 48 50 90 :39
BF40 20 20 00 00 00 00 00 00 :40
BF48 03 1C 00 00 03 0D 31 C1 :21
BF50 01 79 08 04 08 10 20 40 :FE
BF58 20 10 00 00 00 00 00 00 :30
BF60 00 80 00 00 00 00 00 00 :80
BF68 00 01 10 20 10 08 04 02 :4F
BF70 04 08 00 20 50 88 05 02 :0B
BF78 02 07 00 20 50 88 05 02 :08
---------------------------------
SUM: E5 EA 5C FD 88 DB D7 A4 :06

BF80 00 00 00 00 00 00 00 00 :00
BF88 00 01 08 10 20 40 40 40 :F9
BF90 40 50 04 04 09 0A 12 13 :D0
BF98 22 3F 20 A0 90 90 88 08 :D1
BFA0 04 FC 9E 80 83 8C B0 C0 :9D
BFA8 00 00 38 C0 00 00 00 00 :F8
BFB0 00 00 02 02 01 01 01 00 :07
BFB8 00 00 40 40 80 40 40 80 :00
BFC0 80 80 E4 1B 00 00 00 00 :FF
BFC8 00 00 25 1F C1 31 0D 03 :46
BFD0 00 00 09 07 09 10 20 40 :89
BFD8 20 10 00 F8 04 83 00 00 :AF
BFE0 00 00 00 1F 20 C1 00 00 :00
BFE8 00 00 90 E0 90 08 04 02 :0E
BFF0 04 08 0A 02 02 02 02 04 :22
BFF8 08 08 80 00 00 00 00 00 :90
---------------------------------
SUM: 12 2C 70 70 3D 36 FE E4 :73

C000 00 00 00 00 40 A0 11 0A :FB
C008 04 00 E0 40 40 A0 11 0A :1F
C010 04 00 00 00 00 EF 11 2A :2E
```


▶私の妻のお父さん（つまり義父）が，X1turboまたはturboⅡを買うので，今のX1を譲ってもいいと先日，電話をくれました。それで私は今，非常に悩んでいるのです。私は現在MZ-1500を持っているのですが，我家は妻の方針で「一家に一台」しかパソコンを置いてはいけないのです。X1を選ぶとなると，MZ-1500は他の人に譲らないといけないし，

```
C018 2A 11 00 00 00 EF 10 A8 :E2
C020 A8 10 00 1F 10 F3 1C 1F :15
C028 00 00 00 F0 70 9F 10 F0 :FF
C030 00 00 10 28 44 82 01 00 :FF
C038 00 00 10 28 44 82 01 00 :FF
C040 00 00 00 02 02 FE 03 2E :33
C048 30 10 00 4C 70 FF 40 40 :7B
C050 00 00 00 00 00 00 00 01 :01
C058 0F 09 08 14 18 20 40 48 :F4
C060 94 08 08 14 0C 02 01 09 :D0
C068 14 08 00 00 00 00 00 40 :5C
C070 F8 48 05 05 05 FD 05 05 :56
C078 05 00 D0 50 50 5F 50 50 :74
---------------------------------
SUM: BE 92 E5 6A 73 2F 4A 4A :D5

C080 D0 00 00 00 00 00 00 00 :D0
C088 00 00 00 00 00 00 00 00 :00
C090 00 00 00 00 00 00 00 00 :00
C098 00 00 00 00 00 00 00 00 :00
C0A0 00 00 00 00 00 00 00 00 :00
C0A8 00 00 00 00 00 00 00 00 :00
C0B0 00 00 00 00 00 00 00 00 :00
C0B8 00 00 00 00 00 00 00 00 :00
C0C0 00 00 00 00 00 00 00 00 :00
C0C8 00 00 00 00 00 00 00 00 :00
C0D0 00 00 00 00 00 00 00 00 :00
C0D8 00 00 00 00 00 00 00 00 :00
C0E0 00 00 00 00 00 00 00 00 :00
C0E8 00 00 00 00 00 00 00 00 :00
C0F0 00 00 00 00 00 00 00 00 :00
C0F8 00 00 00 00 00 00 00 00 :00
---------------------------------
SUM: D0 00 00 00 00 00 00 00 :D0

C100 00 00 00 00 00 00 00 00 :00
C108 00 00 00 00 00 00 00 00 :00
C110 00 00 38 08 F6 09 06 01 :46
C118 00 00 1C 10 6F 90 60 80 :0B
C120 00 00 38 08 F6 29 26 21 :A6
C128 20 20 1C 10 6F 94 64 84 :57
C130 04 04 F0 08 06 01 00 00 :07
C138 00 00 0F 10 60 80 00 00 :FF
C140 00 00 F0 28 26 21 20 20 :9F
C148 20 20 0F 14 64 84 04 04 :53
C150 04 04 FF 00 00 00 00 00 :07
C158 00 00 FF 00 00 00 00 00 :FF
C160 00 00 FF 20 20 20 20 20 :9F
C168 20 20 FF 04 04 04 04 04 :53
C170 04 04 00 00 00 00 00 00 :08
C178 00 00 00 00 00 00 00 00 :00
---------------------------------
SUM: 6C 6C A2 A8 DE A0 38 6E :46
```

（C180～C1FFは00で埋めてください）

```
C200 00 00 00 00 00 00 00 00 :00
C208 00 00 00 00 00 00 00 00 :00
C210 00 00 00 00 01 06 09 F6 :06
C218 08 38 00 00 80 60 90 6F :1F
C220 10 1C 20 20 21 26 29 F6 :D2
C228 08 38 04 04 84 64 94 6F :33
C230 10 1C 00 00 00 00 01 06 :33
C238 08 F0 00 00 00 00 80 60 :D8
C240 10 0F 20 20 20 20 21 26 :E6
C248 28 F0 04 04 04 04 84 64 :10
C250 14 0F 00 00 00 00 00 00 :23
C258 00 FF 00 00 00 00 00 00 :FF
C260 00 FF 20 20 20 20 20 20 :BF
C268 20 FF 04 04 04 04 04 04 :37
C270 04 FF 00 00 00 00 00 00 :03
C278 00 00 00 00 00 00 00 00 :00
---------------------------------
SUM: A8 A2 6C 6C 6E 38 A0 DE :46
```

（C280～C37Fは00で埋めてください）

```
C380 00 00 00 00 00 00 00 00 :00
C388 00 00 00 00 00 00 00 00 :00
C390 00 00 08 10 20 40 20 10 :A8
C398 08 04 00 00 00 00 00 00 :0C
C3A0 00 00 08 08 68 90 68 08 :78
C3A8 68 90 00 00 00 00 00 00 :F8
C3B0 00 00 08 08 08 3E 32 32 :BA
C3B8 2A 2A 00 00 00 00 00 00 :54
C3C0 00 00 08 08 28 48 C8 28 :70
C3C8 3E 10 00 00 00 00 00 00 :4E
C3D0 00 00 00 00 00 C0 40 40 :40
C3D8 40 E0 00 00 00 00 00 00 :20
C3E0 00 00 00 00 42 A5 46 08 :35
C3E8 30 40 00 00 00 00 00 00 :70
C3F0 00 00 08 08 08 08 7F 00 :9F
C3F8 7F 41 00 00 00 00 00 00 :C0
---------------------------------
SUM: C7 2F 28 30 02 C3 87 BA :54
```

```
C400 00 00 00 00 00 00 00 00 :00
C408 00 00 00 00 00 00 00 00 :00
C410 00 00 08 10 20 40 20 10 :A8
C418 08 04 00 00 00 00 00 00 :0C
C420 00 00 68 08 68 90 68 08 :D8
C428 08 08 00 00 00 00 00 00 :10
C430 00 00 2A 26 26 3E 08 08 :C4
C438 08 08 00 00 00 00 00 00 :10
C440 00 00 08 3E 0C 0C 0A 0A :72
C448 08 08 00 00 00 00 00 00 :10
C450 00 00 40 30 08 46 A5 42 :A5
C458 00 00 00 00 00 00 00 00 :00
C460 00 00 E0 40 40 C0 00 00 :20
C468 00 00 00 00 00 00 00 00 :00
C470 00 00 7F 00 7F 08 08 08 :16
C478 08 08 00 00 00 00 00 00 :10
---------------------------------
SUM: 28 24 41 EC 81 28 47 74 :DD

C480 00 00 00 00 00 00 00 00 :00
C488 00 00 00 00 00 00 00 00 :00
C490 00 00 20 20 BF A0 D0 28 :97
C498 28 14 00 00 00 00 00 00 :3C
C4A0 00 00 04 04 FD 05 0B 14 :29
C4A8 14 28 00 00 00 00 00 00 :3C
C4B0 00 00 80 80 FF 80 40 20 :DF
C4B8 20 10 00 00 00 00 00 00 :30
C4C0 00 00 01 01 FF 01 02 04 :08
C4C8 04 08 00 00 00 00 00 00 :0C
C4D0 00 00 80 80 FF 80 80 80 :7F
C4D8 80 80 00 00 00 00 00 00 :00
C4E0 00 00 01 01 FF 01 01 01 :04
C4E8 01 01 00 00 00 00 00 00 :02
C4F0 00 00 00 00 00 00 00 00 :00
C4F8 00 00 00 00 00 00 00 00 :00
---------------------------------
SUM: E1 D5 26 26 B8 A7 9E E1 :E0

C500 00 00 00 00 00 00 00 00 :00
C508 00 00 00 00 00 00 00 00 :00
C510 00 00 14 28 28 D0 A0 BF :93
C518 20 20 00 00 00 00 00 00 :40
C520 00 00 28 14 14 0B 05 FD :5D
C528 04 04 00 00 00 00 00 00 :08
C530 00 00 10 20 20 40 80 FF :0F
C538 80 80 00 00 00 00 00 00 :00
C540 00 00 08 04 04 02 01 FF :12
C548 01 01 00 00 00 00 00 00 :02
C550 00 00 80 80 80 80 80 FF :7F
C558 80 80 00 00 00 00 00 00 :00
C560 00 00 01 01 01 01 01 FF :04
C568 01 01 00 00 00 00 00 00 :02
C570 00 00 00 00 00 00 00 00 :00
C578 00 00 00 00 00 00 00 00 :00
---------------------------------
SUM: 26 26 D5 E1 E1 9E A7 B8 :E0

C580 00 00 00 00 00 00 00 00 :00
C588 00 00 00 00 00 00 00 00 :00
C590 00 00 20 20 A0 A0 D0 28 :78
C598 28 14 00 00 00 00 00 00 :3C
C5A0 00 00 04 04 05 05 0B 14 :31
C5A8 14 28 00 00 00 00 00 00 :3C
C5B0 00 00 80 80 80 80 40 20 :60
C5B8 20 10 00 00 00 00 00 00 :30
C5C0 00 00 01 01 01 01 02 04 :0A
C5C8 04 08 00 00 00 00 00 00 :0C
C5D0 00 00 80 80 80 80 80 80 :00
C5D8 80 80 00 00 00 00 00 00 :00
C5E0 00 00 01 01 01 01 01 01 :06
C5E8 01 01 00 00 00 00 00 00 :02
C5F0 00 00 00 00 00 00 00 00 :00
C5F8 00 00 00 00 00 00 00 00 :00
---------------------------------
SUM: E1 D5 26 26 A7 A7 9E E1 :CF

C600 00 00 00 00 00 00 00 00 :00
C608 00 00 00 00 00 00 00 00 :00
C610 00 00 14 28 28 D0 A0 A0 :74
C618 20 20 00 00 00 00 00 00 :40
C620 00 00 28 14 14 0B 05 05 :65
C628 04 04 00 00 00 00 00 00 :08
C630 00 00 10 20 20 40 80 80 :90
C638 80 80 00 00 00 00 00 00 :00
C640 00 00 08 04 04 02 01 01 :14
C648 01 01 00 00 00 00 00 00 :02
C650 00 00 80 80 80 80 80 80 :00
C658 80 80 00 00 00 00 00 00 :00
C660 00 00 01 01 01 01 01 01 :06
C668 01 01 00 00 00 00 00 00 :02
C670 00 00 00 00 00 00 00 00 :00
C678 00 00 00 00 00 00 00 00 :00
---------------------------------
SUM: 26 26 D5 E1 E1 9E A7 A7 :CF
```

（C680～C7FFは00で埋めてください）

```
C800 00 00 00 00 00 00 00 00 :00
C808 00 00 00 00 00 00 00 00 :00
C810 00 00 00 18 24 42 7E 42 :3E
C818 42 42 00 7C 22 22 3C 22 :A2
C820 22 7C 00 1C 22 40 40 40 :9C
C828 22 1C 00 78 24 22 22 22 :40
C830 24 78 00 7E 40 40 78 40 :52
C838 40 7E 00 7E 40 40 78 40 :74
C840 40 40 00 1C 22 40 4E 42 :8E
C848 22 1C 00 42 42 42 7E 42 :C4
C850 42 42 00 1C 08 08 08 08 :C0
C858 08 1C 00 0E 04 04 04 04 :42
C860 44 38 00 42 44 48 70 48 :02
C868 44 42 00 40 40 40 40 40 :C6
C870 40 7E 00 42 66 5A 5A 42 :5C
C878 42 42 00 42 62 52 4A 46 :0A
---------------------------------
SUM: A0 C4 00 B2 C8 08 38 E6 :04

C880 42 42 00 18 24 42 42 42 :86
C888 24 18 00 7C 42 42 7C 40 :F8
C890 40 40 00 18 24 42 42 4A :8A
C898 24 1A 00 7C 42 42 7C 48 :02
C8A0 44 42 00 3C 42 40 3C 02 :82
C8A8 42 3C 00 3E 08 08 08 08 :DC
C8B0 08 08 00 42 42 42 42 42 :5A
C8B8 42 3C 00 42 42 42 24 24 :8C
C8C0 18 18 00 42 42 42 5A 5A :AA
C8C8 66 42 00 42 42 24 18 24 :8C
C8D0 42 42 00 22 22 22 1C 08 :0E
C8D8 08 08 00 7E 02 04 18 20 :CC
C8E0 40 7E 00 00 00 38 04 3C :36
C8E8 44 3A 00 40 40 5C 62 42 :FE
C8F0 62 5C 00 00 00 3C 42 40 :7C
C8F8 42 3C 00 02 02 3A 46 42 :44
---------------------------------
SUM: 8A 6A 00 8C 84 6A BA 2A :52

C900 46 3A 00 00 00 3C 42 7E :7C
C908 40 3C 00 0C 12 10 7C 10 :36
C910 10 10 00 00 3A 46 46 3A :20
C918 02 3C 00 40 40 5C 62 42 :BE
C920 42 42 00 08 00 18 08 08 :B4
C928 08 1C 04 00 0C 04 04 04 :40
C930 44 38 00 40 40 44 48 50 :D8
C938 68 44 00 18 08 08 08 08 :E4
C940 08 1C 00 00 00 76 49 49 :2C
C948 49 49 00 00 00 5C 62 42 :92
C950 42 42 00 00 00 3C 42 42 :44
C958 42 3C 00 00 5C 62 62 5C :FA
C960 40 40 00 00 3A 46 46 3A :80
C968 02 02 00 00 00 5C 62 40 :02
C970 40 40 00 00 00 3E 40 3C :3A
C978 02 7C 00 10 10 7C 10 10 :3A
---------------------------------
SUM: E7 7D 04 BC B6 22 09 5D :32

C980 12 0C 00 00 00 42 42 42 :E4
C988 42 3C 00 00 00 42 42 42 :44
C990 24 18 00 00 00 41 49 49 :0F
C998 49 36 00 00 00 44 28 10 :FB
C9A0 28 44 00 00 42 42 46 3A :70
C9A8 02 3C 00 00 00 7E 04 18 :D8
C9B0 20 7E 00 3C 42 42 42 42 :E2
C9B8 42 3C 00 08 18 28 08 08 :D6
C9C0 08 3E 00 3C 42 02 0C 30 :02
C9C8 40 7E 00 3C 42 02 3C 02 :7C
C9D0 42 3C 00 04 0C 14 24 7E :44
C9D8 04 04 00 7E 40 78 04 02 :44
C9E0 44 38 00 1C 20 40 7C 42 :B6
C9E8 42 3C 00 7E 42 04 08 10 :5A
C9F0 10 10 00 3C 42 42 3C 42 :5E
C9F8 42 3C 00 3C 42 42 3E 02 :7E
---------------------------------
SUM: B3 8C 00 50 52 8B F7 C1 :24

CA00 04 38 00 3E 02 3E 02 04 :C0
CA08 08 00 00 00 3E 02 0C 08 :5C
CA10 10 00 00 00 04 08 18 28 :5C
CA18 08 00 00 00 08 3E 22 02 :72
CA20 0C 00 00 00 00 3E 08 08 :5A
CA28 3E 00 00 00 04 3E 0C 14 :A0
CA30 24 00 00 00 10 3E 12 14 :98
CA38 10 00 00 00 00 1C 04 04 :34
CA40 3E 00 00 00 3C 04 3C 04 :BE
CA48 3C 00 00 00 00 2A 2A 02 :92
CA50 0C 00 00 00 00 1C 00 00 :28
CA58 00 00 3E 02 0A 0C 08 08 :66
CA60 10 00 02 04 08 18 28 08 :66
CA68 08 00 08 3E 22 22 02 04 :98
CA70 08 00 00 3E 08 08 08 08 :66
CA78 3E 00 04 3E 04 0C 14 24 :C8
---------------------------------
```

といってMZ-1500を選ぶと義父の好意をムダにしたうえに，X1の豊富なソフトの魅力も捨てなければならないし……。1月号が発売されるまでにこの返事を出さなければならない，こまったもんだ。

細田　茂（27）神奈川県

```
SUM: 86 38 4C FE DC 00 26 B0 :BA

CA80 04 00 10 3E 12 12 12 12 :9A
CA88 24 00 08 3E 08 3E 08 08 :C0
CA90 08 00 00 1E 12 22 02 04 :60
CA98 18 00 10 1E 24 04 04 04 :76
CAA0 08 00 00 3E 02 02 02 02 :4E
CAA8 3E 00 14 3E 14 14 04 08 :C4
CAB0 10 00 00 30 02 32 02 04 :7A
CAB8 38 00 00 3E 02 04 08 14 :98
CAC0 22 00 10 3E 12 14 10 10 :B6
CAC8 0E 00 00 22 22 12 02 04 :6A
CAD0 18 00 00 3C 24 54 0C 08 :E0
CAD8 30 00 04 38 08 3E 08 08 :C2
CAE0 10 00 00 2A 2A 2A 02 04 :94
CAE8 08 00 1C 00 3E 08 08 08 :7A
CAF0 10 00 10 10 10 18 14 10 :7C
CAF8 10 00 08 08 3E 08 08 10 :7E
--------------------------------
SUM: 86 00 84 B8 80 CC 7C 94 :1E

CB00 20 00 00 1C 00 00 00 00 :3C
CB08 3E 00 00 3E 02 14 08 14 :AE
CB10 20 00 08 3E 04 08 1C 2A :B8
CB18 08 00 04 04 04 04 04 08 :24
CB20 10 00 00 08 04 22 22 22 :82
CB28 22 00 20 20 3E 20 20 20 :00
CB30 1E 00 00 3E 02 02 02 04 :66
CB38 18 00 00 10 28 04 02 02 :58
CB40 00 00 08 3E 08 08 2A 2A :AA
CB48 08 00 00 3E 02 02 14 08 :66
CB50 04 00 00 1C 00 1C 00 3C :78
CB58 02 00 00 08 10 20 22 3E :9A
CB60 02 00 00 02 02 14 08 14 :36
CB68 20 00 00 3E 10 3E 10 10 :CC
CB70 0E 00 10 10 3E 12 14 10 :A2
CB78 10 00 00 1C 04 04 04 04 :3C
--------------------------------
SUM: 3C 00 44 1E E4 16 FE 72 :08

CB80 3E 00 00 3E 02 3E 02 02 :C0
CB88 3E 00 1C 00 3E 02 02 04 :A0
CB90 08 00 24 24 24 24 04 08 :A4
CB98 10 00 00 08 28 28 2A 2A :BC
CBA0 2C 00 00 20 20 22 24 28 :DA
CBA8 30 00 00 3E 22 22 22 22 :F6
CBB0 3E 00 00 3E 22 22 02 04 :C6
CBB8 08 00 00 30 00 02 02 04 :40
CBC0 38 00 10 48 20 00 00 00 :B0
CBC8 00 00 70 50 70 00 00 00 :30
CBD0 00 00 00 00 00 00 00 00 :00
CBD8 00 00 00 00 00 00 00 00 :00
CBE0 00 00 00 00 00 00 00 00 :00
CBE8 00 00 00 00 00 00 00 00 :00
CBF0 00 00 00 00 00 00 00 00 :00
CBF8 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: 6E 00 C0 CE 80 F4 7C 8A :76

CC00 00 00 00 00 00 00 00 00 :00
CC08 00 00 00 00 00 00 00 00 :00
CC10 00 00 00 00 00 00 00 3E :3E
CC18 00 00 00 00 00 08 08 3E :4E
CC20 08 08 00 00 00 02 04 08 :1E
CC28 10 20 00 00 00 2A 1C 08 :7E
CC30 1C 2A 00 1C 22 22 22 14 :DC
CC38 14 36 00 00 00 1A 1A 1E :9C
CC40 32 42 00 00 00 00 00 00 :74
CC48 00 00 00 00 00 00 00 00 :00
CC50 00 00 00 00 00 00 00 00 :00
CC58 00 00 00 00 00 00 00 00 :00
CC60 00 00 00 00 00 00 00 00 :00
CC68 00 00 00 00 00 00 00 00 :00
CC70 00 00 00 00 00 00 00 00 :00
CC78 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: 7A CA 00 1C 22 70 64 BE :14

CC80 00 00 00 00 00 00 00 00 :00
CC88 00 00 00 00 00 00 00 00 :00
CC90 00 00 00 00 00 00 00 18 :18
CC98 18 00 00 00 00 00 00 08 :20
CCA0 08 10 00 04 08 10 10 10 :54
CCA8 08 04 00 20 10 08 08 08 :54
CCB0 10 20 00 08 1E 28 1C 0A :A4
CCB8 3C 08 00 00 62 64 08 10 :22
CCC0 26 46 00 30 48 48 30 4A :A6
CCC8 44 3A 00 00 00 3C 00 3C :F6
CCD0 00 00 00 00 00 00 00 00 :00
CCD8 00 00 00 00 00 00 00 00 :00
CCE0 00 00 00 00 00 00 00 00 :00
CCE8 00 00 00 00 00 00 00 00 :00
CCF0 00 00 00 00 00 00 00 00 :00
CCF8 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: DE BC 00 5C E0 28 6C D8 :42

CD00 00 00 00 00 00 00 00 00 :00
CD08 00 00 00 00 00 00 00 00 :00
CD10 00 00 00 04 3F 11 11 12 :77
CD18 0A 7F 00 00 BE 22 24 28 :B5
CD20 28 A4 00 0F 08 08 08 0F :02
CD28 08 00 00 F0 10 10 10 F0 :18
CD30 10 00 00 10 3F 48 04 04 :AF
CD38 7F 0E 00 10 3F 44 08 7E :A6
CD40 42 42 00 00 00 00 00 00 :84
CD48 00 00 00 00 00 00 00 00 :00
CD50 00 00 00 00 00 00 00 00 :00
CD58 00 00 00 00 00 00 00 00 :00
CD60 00 00 00 00 00 00 00 00 :00
CD68 00 00 00 00 00 00 00 00 :00
CD70 00 00 00 00 00 00 00 00 :00
CD78 00 00 00 00 00 00 00 00 :00
--------------------------------
SUM: 0B 73 00 23 93 D7 59 BB :1F

CD80 00 00 00 00 00 00 00 00 :00
CD88 00 00 00 00 00 00 00 00 :00
CD90 00 00 00 3F 21 21 21 3F :E1
CD98 21 00 22 22 2C 20 20 20 :F1
CDA0 20 00 00 7E 42 42 42 7E :E2
CDA8 42 00 00 7E 42 42 42 7E :04
CDB0 42 00 0E 15 15 24 44 04 :E6
CDB8 04 00 7E 42 42 7E 42 42 :08
CDC0 7E 00 00 78 44 79 64 78 :8F
CDC8 01 00 00 07 0C 47 C3 8F :AD
CDD0 00 00 00 3D 19 19 99 18 :20
CDD8 00 00 00 37 34 34 35 E7 :BB
CDE0 00 00 00 9C 49 C9 DD 9C :27
CDE8 00 00 00 E2 32 70 70 E0 :D4
CDF0 00 00 00 73 82 73 3A F2 :94
CDF8 00 00 00 96 5B D3 D3 D3 :6A
--------------------------------
SUM: 48 00 AE 2E 1D F3 9A E8 :B6

CE00 00 00 00 7B 32 33 32 32 :44
CE08 00 00 00 80 40 C0 C8 C8 :10
CE10 00 00 FF 80 AF AF A0 BF :3C
CE18 BE BD FF 01 F5 F5 05 FD :67
CE20 7D BD FF FF E0 EF EF E8 :DE
CE28 EF E8 FF FF 07 F7 F7 17 :E1
CE30 F7 17 FF 80 BF BF B5 BF :7F
CE38 B5 BE FF 1F DF DF 4B DF :79
CE40 5B 5F FF 00 AA 00 55 00 :B8
CE48 AA 00 FF 0F AF 0F 6F 0F :F4
CE50 AF 0F FF FF FF FF FE FD :B5
CE58 FA F1 FF DF AF 57 B3 6D :EF
CE60 DB B7 FF BB 92 AA BA BA :FC
CE68 FF C0 FF 63 AD 23 AF AF :4F
CE70 FF 03 FF F0 F7 F4 F6 E4 :B6
CE78 D6 D7 FF 0F EF 6F 2F 67 :AF
--------------------------------
SUM: 33 E7 F2 23 C7 B0 88 80 :AE

CE80 2B EB FF FF FF FF FF FF :10
CE88 FF FF FF FF FF FF FF FF :F8
CE90 FF FF BE BF BE BE BE BF :74
CE98 80 FF 7D FD 7D 7D 7D FD :6D
CEA0 01 FF E8 EF EF E0 E0 F0 :76
CEA8 FF FF 17 F7 F7 07 07 0F :20
CEB0 FF FF B4 BF 80 F7 FF F5 :DC
CEB8 FF FF 5B DF 1B FF FB 53 :A0
CEC0 FF FF FF D7 DB E1 FB F7 :82
CEC8 FF FF FF B3 55 13 55 53 :C0
CED0 FF FF F5 EE EF DE D9 87 :0E
CED8 9C FF 6F DF 3F 7F FF FF :A5
CEE0 01 FF DF DA D2 D8 D2 DF :14
CEE8 C0 FF FB 5B 8B 5B AB FB :A1
CEF0 03 FF B0 BF 80 BF BF BF :2E
CEF8 80 FF 0D FD 01 FD C5 FD :49
--------------------------------
SUM: 84 DC 40 86 F6 56 43 67 :1C

CF00 01 FF FF FF FF FF FF FF :FA
CF08 FF FF FF FF FF FF FF FF :F8
CF10 FF FF FF 81 BF A0 A0 A0 :1D
CF18 A0 E0 07 05 05 05 05 FD :98
CF20 81 FF 00 00 00 00 00 00 :80
CF28 00 00 00 00 00 00 00 00 :00
CF30 00 00 00 00 00 00 00 00 :00
CF38 00 00 00 00 00 00 00 00 :00
CF40 00 00 00 00 00 00 00 00 :00
CF48 00 00 00 00 00 00 00 00 :00
CF50 00 00                   :00
--------------------------------
SUM: 20 DC 04 84 C2 A3 A3 9B :27
```

リスト4 SUB.OBJ

```
C600 C3 1E C6 C3 2F C6 C3 5E :80
C608 C6 C3 3F C6 C3 73 C6 C3 :4D
C610 51 C7 C3 0A C7 C3 58 C7 :8E
C618 C3 8E C7 C3 9D C7 CD DE :EA
C620 C6 3A F2 CF 77 11 70 17 :D0
C628 19 3A F3 CF 77 18 44 CD :B5
C630 DE C6 7E 32 F2 CF 11 70 :96
C638 17 19 7E 32 F3 CF C9 DF :4A
C640 2D 3E 06 32 00 10 21 E0 :B4
C648 2E 22 14 10 21 00 D0 22 :87
C650 16 10 21 00 00 22 18 10 :91
C658 11 00 D0 DF 31 C9 DF 2D :C6
C660 DF 2F 3A 00 10 FE 06 20 :7C
C668 06 2A 16 10 DF 30 C9 3E :6C
C670 3D DD E9 CD F4 C6 F3 11 :8E
C678 C5 C7 D9 06 18 D9 01 18 :75
--------------------------------
SUM: DA F6 8D 5C 76 52 E7 BF :27

C680 00 ED B0 E5 EB 11 10 00 :8E
C688 19 EB E1 D5 11 4C 00 19 :30
C690 D1 D9 05 28 03 D9 18 E6 :B1
C698 D9 CD F4 C6 11 70 17 19 :11
C6A0 11 85 CB D9 06 18 D9 01 :32
C6A8 18 00 ED B0 E5 EB 11 10 :A6
C6B0 00 19 EB E1 D5 11 4C 00 :17
C6B8 19 D1 D9 05 28 03 D9 18 :E4
C6C0 E6 D9 D3 E3 11 28 D4 21 :A3
C6C8 C5 C7 01 C0 03 ED B0 11 :FE
C6D0 28 DC 21 85 CB 01 C0 03 :39
C6D8 ED B0 D3 E1 FB C9 3A F1 :40
C6E0 CF 47 11 64 00 21 9C CF :17
C6E8 19 10 FD 16 00 3A F0 CF :35
C6F0 3D 5F 19 C9 3A F6 CF 47 :C4
C6F8 11 64 00 21 9C CF 19 10 :2A
--------------------------------
SUM: FB 33 F5 84 A8 BC 40 5C :A7

C700 FD 16 00 3A F5 CF 3D 5F :AD
C708 19 C9 F3 3E 02 D3 E5 11 :DE
C710 00 20 21 00 D0 3E 00 3E :8D
C718 00 77 23 1B 7A B3 20 F7 :F9
C720 D3 E6 D9 21 3C 00 11 3C :3C
C728 00 D9 21 6F E7 06 64 0E :C8
C730 3C 23 7E CB 5F D9 28 04 :0C
C738 3E 01 DF 48 13 D9 05 28 :7F
C740 04 18 EE 18 08 06 64 D9 :6D
C748 23 11 3C 00 D9 0D 20 E1 :57
C750 C9 AF DF 0D 32 F4 CF C9 :22
C758 21 9D CB 06 10 0E 10 11 :CE
C760 18 00 3A F3 CF 77 23 05 :B3
C768 20 F8 0D 28 05 19 06 10 :81
C770 18 F0 21 DD C7 06 10 0E :F1
C778 10 11 18 00 AF 77 23 3C :BE
--------------------------------
SUM: D4 C7 E2 59 43 6D A3 0E :37

C780 05 20 FA 0D 28 05 19 06 :78
C788 10 18 F2 C3 C2 C6 11 E0 :56
C790 2E 21 00 D0 AF 77 23 1B :83
C798 7A B3 20 F8 C9 62 6B 7E :59
C7A0 23 CD A7 C7 10 F9 C9 0E :3E
C7A8 00 57 CD BD C7 7A D3 FF :F4
C7B0 3E 80 D3 FE 0E 01 CD BD :28
C7B8 C7 AF D3 FE C9 DB FE E6 :CF
C7C0 0D B9 C8 18 F8 00 00 00 :9E
--------------------------------
SUM: F2 18 EE 30 08 F3 1F 2F :71
```

気分をマルチにパワーアップ

小澤設計・計画室
Ozawa Sho
小澤　尚

連載7回目をむかえ，最終回となりました。部品化のプログラムを交え，その他のちょっとした付属プログラムを紹介しましょう。

あなたの好みに向けて

平面的に入力した図が3次元の図に出力するこのシリーズは，大仰にも“マルチ”という形容詞がついています。たかだかワイヤーフレームやソリッドモデルの透視図を出力するのに省入力化したに過ぎないと思っておられる方もいるでしょう。確かに今まで紹介した内容の範囲は，プレゼンテーション（表現）用の道具の範囲で，まだ真のマルチにはほど遠いものかもしれません。CADという略語も，ものの表現にとどまらない意味があります。

今回紹介する「部品化のプログラム」や「通信用のプログラム」は，そのマルチ化の入口として考えているものですが，さらに向上を望むならば，データベースとして種種の計算や，構造上あるいは機能上のシミュレーションチェックが必要となるところかもしれません。しかし，ここから先になると分野や利用方法によってさまざま，特殊化の道に入らねばならないことでしょう。要は，いかにあなた好みにできるかなのだと思いますが，ここから先は，ご自身の工夫を期待し，真のマルチCADにふくらませていただければと望んでいます。

そんな参考として，今回の「部品化プログラム」を位置付けてほしいと考えているのですが……。

部品化のプログラム

リスト1が，マルチCADで作図したものを部品化し，登録・呼び出しを行えるプログラムです。このリストは8月号に載った入力作図用プログラム“MAIN-JOB”(MAIN-JOBは，リスト3の改造をしてください) MERGEしてできるようになっています。したがって，このリストを入力したらアスキー形式でセーブ（ファイル名の後に，Aをつける）し，“MAIN-JOB”をロードしてから，このファイルをMERGEのコマンドで呼び出すと，部品化のプログラムはできあがります。このファイル名は，何でもかまいませんが，9月号の“JOB SELECT”と整合させれば“トクシュニュウリョク”となります。

さてこの操作方法は8月号の入力プログラムと同様ですので，新しく加わった機能のみ説明しましょう。新しい機能は画面右側にPLO.（部品呼び出し），PSA.（部品登録），テイセイの3つが加わっただけですので簡単です。これ以外は入力用のプログラムと同様ですので，絵を描く機能ももちろんあります。登録したい部品の入力はこの部品用プログラムでも8月号の入力用プログラムでも，あるいは既存データファイル呼び出してもかまいません。

さて，操作は部品にしたい図が入力されたらカーソルを「PSA.」の欄にもっていき指定すると，画面下に位置指定の表示がでます。そこで，登録したい図の範囲にカーソル⊞を移動させ，左上，右下の順に指定します（この範囲は，少し広めにとっておいてください)。すると，既存データのファイルが表示され，ファイル名記入の入力待ちになります。ファイル名は，入力された名の前に必ず「P-」がつくようになっています。ファイル名を入力すると計算が始まり，しばらくするとデータファイルに登録され，入力した図が消えます。これで登録は完了し，その他の入力待ちになります(図1～2)。部品の呼び出しはPLO.を指定し，操作手順は登録と同様，記入したい位置の範囲を指定し，データファイル名を入力します。部品のデータファイルは必ず「P-」が頭についていますので，「P-」つきのファイル名となります。ファイル名を入力すると，ロードに続いて計算が始まり，指定した位置と範囲に作図され(図3)，次の入力待ちとなります。呼び出されて記入された図の奥行きの倍率（3次元に起こすためのもの－8月号，9月号参照）は，(部品登録時の倍率）×（呼び出し時に指定されている倍率）となっていますので，3次元に起こしたい図であれば，部品を呼び出す前に，適切な奥行設定（F8キー）をしておくほうが良いでしょう（図4～5）。

図1　部品登録

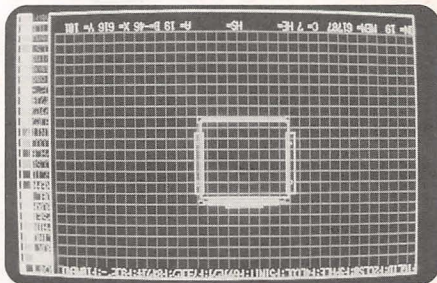



図2　登録

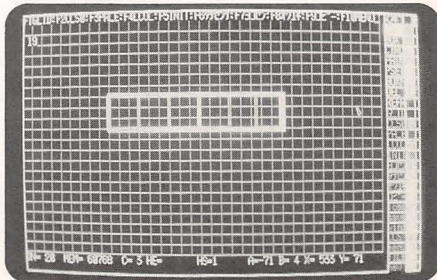



図3　呼び出し合成

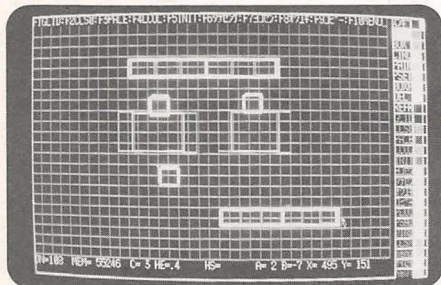



図4　アクソメ・ソリッド出力

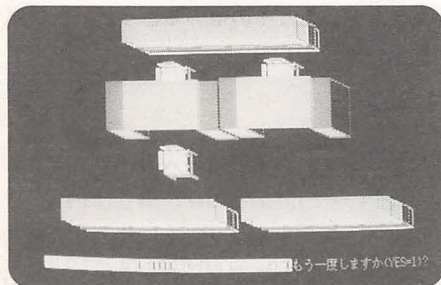
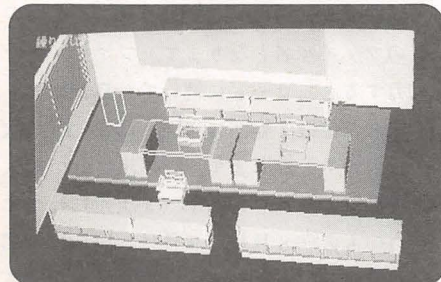



図5　ソリッド・パース出力

さて，これらの部品機能は単によく使う図を，熟語的に使えるだけでなく，違うスケールのものを呼び出し合成することにも使えますので結構便利だと思います。室内のレイアウトであれば，家具などの部品登録といった使われ方がすぐ思い浮かぶところでしょう。

3つ目の機能「テイセイ」は，図の中で不要部分がある場合，その指定範囲をカーソルで左上，右下の対角線位置に指定すると，その部分の作図データを削除する機能です。これが実行されると，指定された部分を除いてデータがソートされ，図も削除されたものが出力されます。2つ以上の作図要素(線など)を一度に指定してもすべてが削除されないことがありますので，その場合は再度「テイセイ」を呼び，残された削除部分を指定してください。この削除機能は，作図コマンドにある「DEL」に比べると任意のものが消去できますので便利に使えるのではないでしょうか。

以上がこの部品化プログラムの説明ですが，このプログラムファイルはファイル名のごとく，いろいろな工夫が付け加えられるように用意しようと思ったものです。

デジタイザがつながるような機能をつけるとか，図を回転させたデータに変換する機能とかいろいろあると思います。貴方自身の工夫も施してみてください。

その他のプログラムファイル

今までいろいろなプログラムファイルを紹介しましたが，3D出力した図をカラープリンタに出力するとか，グラフィック画面自体をセーブするとか，まだメニューの中にあって紹介していないものが残っています。しかし，これらは，いずれも短く，内容も簡単なもの。紹介するまでもないかと思います。

リスト2も，そんな中のひとつであり，データ交信という意義にかこつけた付属ファイルです。これは，X1turbo用ですが，違う機種間でのデータのやりとりや，離れたところのデータ交信に役立てばという願いから作ったもので，参考までにとあえて紹介しました。操作方法は両機種の送受信タイミングを合わせるくらいといったところでしょうか。

ゲーム・ビジネス論

いよいよエンディング，ここはビシッと決めたいところ。さて，どうなることやら！

さて，パソコン雑誌のおおかたは，ホビー(ゲーム)とビジネスに区分されています。多分便宜的な区分だと思いますが，パソコンは確実にビジネスの戦力として歩んでいるようです。人間を管理したり，脅かしにパソコンが使われるのを危惧し,「パソコンは,ゲームにとどまっているほうがましだ」という意見があります。端的にとらえている言葉とは思いますが，こう言ったところで確実にビジネスユースは進んでおり，犬の遠吠えとして無視されかねません。やはり，もう一歩踏み出したビジョンが欲しいのではないでしょうか。

ビジネスとされている分野もどこかにゲーム的な要素を持っているのではないでしょうか。事業を起こすことは，ある意味でバクチにたとえられます。建築の設計も建物と人間をあれこれ考えるゲームととらえることもできるでしょう。またここに紹介した3DマルチCADは，遊びとしての利用もあるはずです。今の世は，遊びが仕事を生む，あるいは，芸術家のように遊び的なものを仕事としている人もいます。これからの仕事は創造性を求められることが多くなるでしょうから，遊びの要素はもっと拡大していくかもしれません。ひょっとしたら，ゲームとビジネスの違いはなくなってしまうのかもしれません。

新聞で，人生のゲームを3分類した記事を見たことがあります。確か，パワー(力)，ゲーム，ウェル(富)ゲーム，ノリッジ(知)ゲームに分類した内容だったと憶えています。もしも「力のゲーム」が，柔道や野球のようなスポーツであれば，自らを磨く素晴しいゲームですが，権力の争奪や，一般市民をまきこむ戦争ならば，問題は大きいでしょう。

「富のゲーム」は，自らそして子孫も含めた社会資本の蓄積に役立つものであればありがたいのですが，詐欺もどきのマネーゲームではいただけません。かたや「知のゲーム」が，真理を見つめようとする目的や，自らを探る目的であれば，素晴しいことだと思いますが，他人を辱めたり，陥れるならば，罪なものと思います。

現実には，ビジネスと称されているものにはいろいろな要素がからんでくるので，この3分類×2分類では，単純には区分できないかもしれません。特に善悪の評価などなかなかつけにくいものでしょう。良かれと思って行動したことが，裏目に出ることだってあるのですから。しかし，事前に心がけを，事後に反省を行うことぐらいはできそうです。

余談ですが，ビジネスという言葉は，busy(忙しい)と名詞化するnessが，活用変化されたつづり busi-nessでできています。一般的に，仕事は忙しい方が良いとされていますが，この“忙”という文字は，忄と亡が組み合わさっています。忄は，心を意味しますから，「心を亡くす」あるいは「亡くす心」となります。とすると，先人の意味した「忙」とは，どうも良い言葉ではなさそうです。世の中，ビジネスと称した悪の所業が結構はびこっているように思えますが「忙」という文字を作った真意は，こうしたことを見込んだことにあるのでしょうか？

ハート・ゲーム

他人と差をつけたいと思う消費者の心理をくすぐる商品が，徐々に多くなっている昨今かと思います。デザインがもてはやされるのは，こうした風潮も手伝っているのでしょう。他人より目立ちたい，立派に見せたい，笑われたくないがためのデザインを選ぶ。これが新たな商品を呼び，エスカレートさせていく――こんな構図も頭に浮かんできます。

また，より刺激的なものを求める心がより刺激的なものを生み出していく，こんな構図も浮かぶでしょう。ここには，提供する側も，される側にも誘惑があります。

デザインをこの構図の中に位置づける人がいるかもしれません。しかし，一方ではデザインを「本質の設計を求めること」と，とらえようとする考えもあります。よりよいデザインとは，「デザインしたものがとり立てて目立ってはいけない」，「飾り立ててはいけない」という考えです。「本質を求める」というのは，自らの理解の浅さや矛盾をはらんでいますから，もちろんデザインされたものが本質にはならないでしょう。しかしそうした目を養う，あるいは作ろうとする心がけはできるかもしれません。

さて，話は何やら抽象的になってきましたが，最後ということもあって肩に力がはいり過ぎてしまったのでしょう。

世の変化は激しく，先は読めなくどこに価値を置いていいかわからない昨今，ビジネスゲームに明け暮れるのが精一杯といった感もありますが，せめて心がけだけでも燃やし続けられる大きな目標を持ちたいと思います。

突然で笑われるかもしれませんが，近頃その指標として，愛があるか，積み上げがある行為かを考えてみてはと思っています。(実は，私の長女は「愛」，次女は土を積み上げる意味で「圭」と名付けたこじつけもあったりしているのですが……)。

パソコンと接して始まることは，自らのプログラミングの旅だと思います。そこには，当面，何をさせるか自ら決め，自ら築いていかねばならぬ厳しさがあります。

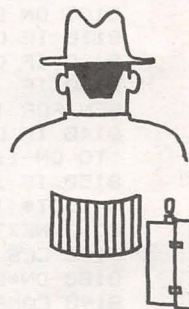
特に難しいのは，何をさせるかで，特に問われるのは，そのはるか向こうに何を願うかでしょう。あるいは，その人の立脚点かもしれません。もう，ここまで考えると，自らの生き方にあるのかもしれませんね。人生のプログラミングを語る資格はありませんが，どこか人とのふれ合いも，そのプログラミングのソースとしてあるような気がしてきます。

おわりに

連載の終盤は，かっこ良く決めてやろうと力んでみたものの，筆はにぶるばかり。ハチャメチャなパソコン論まで出る始末。

あたたかく見守ってくれる読者の皆さんに甘えられたからこそできたのです。

連載が決まったときY氏にいわれた最初の命題「あるときは子供のように，そして あるときは大人の顔を」が今でも頭から離れません。それは，素朴な夢，大きな夢をどう見て，実現してゆくのが問われているように思えました。しかし，実力以上のことはできるはずもなく，浮き足し立ちと持ち前の臭味で，ご覧のとおり。

でも，また再び自らの非力を省みず，より良い環境作りの蓄積を夢見ることにします。どうも長い間ありがとうございました。

連続する線を描くLINE文に注意

NEW BASICおよびturbo BASICの場合，前にXORなどのグラフィック文があるときは連続線の頂点が表示されません。BASICマニュアルには書かれていませんが，座標値の後にPSET，Cを指定すると解決します。アクソメソリッドをNEWまたはturbo BASICで利用する場合は400行などに注意してください。

リスト1　部品登録プログラム

```
5 '*********************************************************************************
10 'ファイルメイ "トクシュショウケン"----ブヒントウロク  プログラム-----            FOR X1
15 '*********************************************************************************
20 IF KAI=0 THEN KAI=1
30 'ON ERROR GOTO40000
40 RESTORE 50:FOR I=0 TO 8:READ B$(I):NEXTI
50 DATA  ズンポ°,LINE,BOX ,CIRC,FAIN,PSET,BOXF,DEL ,REFR
60 RESTORE 70:FOR I=0 TO 8:READ B1$(I):NEXTI
70 DATA  GLID ,CLS0,PALE,LCOL,INIT,ヨコセン,タテセン,オクユキ ,コピ°-
80 RESTORE 90:FOR I=0 TO 6:READ B2$(I):NEXTI
90 DATA  PLO.,PSA.,WIND,CLS ,MENU,テイセイ,NEW
100 IF MAU<>1 THEN X=319:Y=99:GOTO140 ELSE LOADM"BMOUS"
260 ON KEY GOSUB 6000,6120,6200,6300,6400,6520,6620,6800,6900,2040
1900 '*************************
1910 IF E<18 THEN F1=E-8:GOTO1930
1920 F2=E-17:GOTO2030
1930 ON F1 GOSUB 6000,6120,6200,6300,6400,6500,6600,6800,6900
1940 GOTO230
1950 '
1960 '
1970 '
1980 '
1990 '
2000 '
```

```
2010 '
2020 '
2030 ON F2 GOTO 9000,9500,2600,2070,2040,8000,2050
8000 '--------ﾃｲｾｲ-----
8010 BEEP:BEEP:CONSOLE23,2:CFLASH1:PRINT"ｼｮｳｷｮ ﾌﾞﾌﾞﾝ ｻﾞﾋｮｳ ﾉ ｻｲｼｮﾉ ｼﾃｲ ｾﾖ":CFLAS
H0:BO=0
8020 GOSUB 5000:BEEP:CLS
8030 CLS:X1=X:Y1=Y:PRINT"ｼﾃﾝ  X1,Y1=";X,Y,:PRINT"ｼｭｳﾃﾝ ｼﾃｲｾﾖ":BO=1
8040 GOSUB 5000:BEEP:CLS
8050 X2=X:Y2=Y:PRINT"ｼｭｳﾃﾝX2,Y2=";X,Y
8060 IF X2-X1<0 THEN SWAP X1,X2
8070 IF Y2-Y1<0 THEN SWAP Y1,Y2
8075 CFLASH1:PRINT"ｼﾃｲ ﾌﾞﾌﾞﾝ ｼｮｳｷｮ ｼｮﾘ ﾁｭｳ";:CFLASH0
8080 FOR J=0 TO DN-1
8090 D$=DT$(J)
8100 C1=ASC(MID$(D$,1,1))-&H40:C2=VAL(MID$(D$,2,3)):C3=VAL(MID$(D$,6,3)):C4=VAL(
MID$(D$,10,3)):C5=VAL(MID$(D$,14,3)):CL$=MID$(D$,19,1):C=VAL(CL$):IF CL$="" OR C
>7 THEN C=7
8105 ON C1 GOTO 8110,8110,8110,8140,8150,8150,8110
8110 IF C2>C4 THEN SWAP C2,C4
8120 IF C3>C5 THEN SWAP C3,C5
8130 IF (C1=1 OR C1=2 OR C1=3 OR C1=7) AND C2>X1 AND C4<X2 AND C3>Y1 AND C5<Y2 T
HEN FOR I=J TO DN-1:DT$(I)=DT$(I+1):NEXTI:HIKU=HIKU+1
8140 IF C1=4 AND C2>X1 AND C2<X2 AND C3>Y1 AND C3<Y2 AND 4*C4<X2-X1 THEN FOR I=J
 TO DN-1:DT$(I)=DT$(I+1):NEXTI:HIKU=HIKU+1
8150 IF (C1=5 OR C1=6) AND C2>X1 AND C2<X2 AND C3>Y1 AND C3<Y2 THEN FOR I=J TO D
N-1:DT$(I)=DT$(I+1):NEXTI:HIKU=HIKU+1
8160 NEXTJ
8170 CLS
8180 DN=DN-HIKU:HIKU=0:LI=0:BO=0:EBF=0
8190 CONSOLE:GOSUB5610:GOSUB "I":GOSUB5600:GOTO 230
9000 'ﾌﾞﾋﾝ
9010 BEEP:BEEP:CONSOLE23,2:PRINT"ﾌﾞﾋﾝ ﾖﾐｺﾐ ｲﾁｼﾃｲ":BO=0
9020 GOSUB 5000:X1=X:Y1=Y:BO=1:GOSUB 5000:BO=0
9030 IF X1>X THEN SWAP X1,X
9040 IF Y1>Y THEN SWAP Y1,Y
9050 GOSUB 3000:CONSOLE:CLS
9060 FILES"1:":CONSOLE0,25,40,40:CLS
9070 INPUT"ﾖﾐｺﾐ ﾌﾞﾋﾝ ﾉ ﾌｧｲﾙ ﾒｲ ｦ ｲﾚﾃｸﾀﾞｻｲ";PFI$
9075 CFLASH1:PRINT"ﾌﾞﾋﾝ ﾖﾐｺﾐ ｼｮﾘ ﾁｭｳ";:CFLASH0
9080 OPEN"I",#1,"1:"+PFI$
9090 INPUT#1,X2,Y2,X3,Y3
9100 KX=(X-X1)/(X3-X2):KY=(Y-Y1)/(Y3-Y2)
9110 INPUT#1,LN:Z=0:ZS=0:EN=DN+LN
9120 FOR I=DN TO EN
9130 LINPUT#1,DT$:DT$(I)=DT$ D$=DT$(I)
9140 KMZS$=MID$(D$,23,3):KMZS=VAL(KMZS$):IF ZS>KMZS THEN ZS=KMZS
9150 KMZ$=MID$(D$,20,3):IF VAL(KMZ$)=0 THEN KMZ=1 ELSE KMZ=VAL(KMZ$)
9160 IF Z<KMZ THEN Z=KMZ
9170 NEXTI:CLOSE
9180 HSOK=VAL(HSOK$):IF VAL(HOK$)=0 THEN HOK=1 ELSE HOK=VAL(HOK$)
9190 ZK=(HOK-HSOK)/(Z-ZS)
9200 FOR I=DN TO EN:D$=DT$(I):E$=MID$(D$,1,1)
9210 C2=VAL(MID$(D$,2,3)):C3=VAL(MID$(D$,6,3)):C4=VAL(MID$(D$,10,3)):C5=VAL(MID$
(D$,14,3)):CL$=MID$(D$,19,1):C=VAL(CL$):IF CL$="" OR C>7 OR C<1 THEN C=7
9220 KMZS$=MID$(D$,23,3):KMZS=VAL(KMZS$)
9230 KMZ$=MID$(D$,20,3):IF VAL(KMZ$)=0 THEN KMZ=1 ELSE KMZ=VAL(KMZ$)
9240 HSK=HSOK+KMZS*ZK:HK=HSOK+KMZ*ZK:HK$=MID$(STR$(HK)+"    ",2,3):HSK$=MID$(STR$
(HSK)+"    ",2,3)
9250 C2=INT(C2*KX+X1):C3=INT(C3*KY+Y1)
9260 IF E$="A" OR E$="B" OR E$="C" OR E$="G" OR E$="W" THEN C4=INT(C4*KX+X1):C5=
INT(C5*KY+Y1)
9265 IF E$="D" THEN C4=INT(C4*KY)
9268 IF E$="A" THEN DT$(I)=E$+MID$(STR$(C2)+"    ",2,3)+","+MID$(STR$(C3)+"    ",2
,3)+","+MID$(STR$(C4)+"    ",2,3)+","+MID$(STR$(C5)+"    ",2,3)+ ","+MID$(D$,18,8)
:GOTO9280
9270 DT$(I)=E$+MID$(STR$(C2)+"    ",2,3)+","+MID$(STR$(C3)+"    ",2,3)+","+MID$(ST
R$(C4)+"    ",2,3)+","+MID$(STR$(C5)+"    ",2,3)+ ","+STR$(C)+HK$+HSK$
9280 NEXTI:PALET:CONSOLE:CLOSE:CLS
9290 GOTO 2290
9500 'ﾌﾞﾋﾝ ﾄｳﾛｸ-------
9510 BEEP:BEEP:LOCATE0,23:PRINT"ﾌﾞﾋﾝ ﾄｳﾛｸ ｲﾁｼﾃｲ":BO=0:LOCATE20,23:PRINT"ｼﾃﾝ";
9520 GOSUB 5000:X1=X:Y1=Y:BO=1:GOSUB 5000:BO=0
9530 IF X1>X THEN SWAP X1,X
```


▶'85年3月号のMUSIC EDITERを，やっとの思いで入力し終わりました。現在，中学時代の音楽の教科書を引っぱり出してきて悪戦苦闘しているところです。
加藤　一隆（17）愛知県

```
9540 IF Y1>Y THEN SWAP Y1,Y
9550 GOSUB 3000:CLS
9560 FILES"1:":CONSOLE0,25,40,40:CLS
9570 INPUT"トウロク   ブ゛ヒン ノ ファイル メイ ヲ イレテクダ゛サイ  P-";PFI$:PFI$="P-"+PFI$
9575 CFLASH1:PRINT"ブ゛ヒン トウロクチュウ";:CFLASH0
9580 OPEN"O",#1,"1:"+PFI$
9590 PRINT#1,X1,Y1,X,Y
9600 PRINT#1,DN
9610 FOR I=0 TO DN
9620 D$=DT$(I):E$=MID$(D$,1,1)
9630 C2=VAL(MID$(D$,2,3)):C3=VAL(MID$(D$,6,3)):C4=VAL(MID$(D$,10,3)):C5=VAL(MID$
(D$,14,3)):CL$=MID$(D$,19,1):C=VAL(CL$):IF CL$="" OR C>7 THEN C=7
9640 KMZ$=MID$(D$,20,3)
9650 KMZS$=MID$(D$,23,3)
9660 C2=C2-X1:C3=C3-Y1
9670 IF E$="A" OR E$="B" OR E$="C" OR E$="G" OR E$="W" THEN C4=C4-X1:C5=C5-Y1
9680 DT$(I)=E$+RIGHT$("   "+STR$(C2),3)+","+RIGHT$("   "+STR$(C3),3)+","+RIGHT$(
"   "+STR$(C4),3)+","+RIGHT$("   "+STR$(C5),3)+","+STR$(C)+RIGHT$("   "+KMZ$,3)+
KMZS$
9685 IF E$="A" THEN DT$(I)=E$+MID$(STR$(C2)+"   ",2,3)+","+MID$(STR$(C3)+"   ",2
,3)+","+MID$(STR$(C4)+"   ",2,3)+","+MID$(STR$(C5)+"   ",2,3)+ ","+MID$(D$,18,8)
9690 PRINT#1,DT$(I)
9700 NEXTI:IF GL=1 THEN GL=0:GOSUB6000
9710 CLOSE:PALET:CONSOLE:GOSUB5610:CLS0:GOSUB5600:DN=0:GOTO 100
```

リスト2 通信用プログラム

```
5 '**************************************************************************
10 '---ファイルメイ  "ツウシン"---RS-232C-----------FOR X1 TURBO
11 '**************************************************************************
14 CLOSE
15 'ON ERROR GOTO40000
16 ON KEY GOSUB30000,30000,30000,30000,30000,30000,30000,30000,30000
17 CONSOLE0,25:CLS:LOCATE0,24:PRINT"F1-MENU※2-MENU※3-MENU※4-MENU※5-MENU※6-MENU※7-MENU※8-MENU※9-MENU※10CONT
20 GOSUB3010:CONSOLE 5,15,20,40:COLOR2:KMODE1
30 PRINT"データ通信 RS-232C  300ホ゛-  8ヒ゛ット"
40 PRINT:COLOR4
100 PRINT"1:DATA IN -受信"
120 PRINT"2:DATA OUT-送信"
128 PRINT"3:MENU-システムメニューにもどる "
129 PRINT:COLOR1
130 INPUT"どれを行なうか番号を選んで下さい";YN
140 IF YN=1 THEN GOTO"DLOAD"
150 IF YN=2 THEN GOTO"DSAVE"
165 IF YN=3 THEN INIT:CHAIN"JOB SELECT"
170 GOTO10
2190 LABEL"DLOAD"
2200 GOSUB3010:CONSOLE:PRINT" DATA LOAD-RS232C を確認してください":PRINT" 今度呼びだすデ゛ータと合成しますか(NO=0  YES=1)";
:INPUT MA
2210 GOSUB3000
2220 OPEN"I",#1,"COM:1N83XNLLJD"
2225 'FOR I=0 TO 24:LINPUT #1,TA$(I):PA$(I)=HEXCHR$(TA$(I)):NEXTI
2230 LINPUT#1,LN$:LN=VAL(LN$):IF MA=1 THEN EN=LN+DN:ST=DN ELSE ST=0 EN=LN
2235 FOR I=ST TO EN
2240 LINPUT#1,DT$
2245 PRINTI,DT$
2247 DT$(I)=DT$
2250 NEXTI
2252 CLS:CLOSE
2255 DN=EN
2260 GOTO10
2290 LABEL"DSAVE"
2300 GOSUB3010:INPUT "DATA OUT-RS232C を確認してください";YN
2320 OPEN"O",#1,"COM:1N83XNLLJD"
2325 'FOR I=0 TO 24:PRINT#1,TA$(I):NEXTI
2330 PRINT#1,DN:FOR I=0 TO DN
2340 DT$=DT$(I):PRINT#1,DT$
2345 PRINTI,DT$
2350 NEXTI:CLS:CLOSE
2360 INPUT"イママデ゛ノ デ゛ータ ハ ナクシ マスカ(YES=1 NO=2)";YS:IF YS=1 THEN DN=0
```

▶全国のMZ-2000ユーザーの皆さん！ 11月17日の「天才たけしの元気が出るテレビ」にMZ-2000が登場しました。ご出演，おめでとうございます。 但馬 稔郎 (15) 福島県

```
2370 GOTO10
3000 FOR I=0 TO 7:PALETI,I:NEXT:RETURN
3010 FOR I=0 TO 7:PALETI,0:NEXT:RETURN
30000 CHAIN"JOB-SELECT"
40000 '------
40005 INIT:FOR I=7 TO 2 STEP-1:COLOR7,I:NEXTI:CFLASH1
40010 IF ERR=73 THEN PRINT" ｼｭｳﾍﾝ ｷｷ ｶﾞﾂﾅｶﾞｯﾃ ｲﾏｾﾝ":PLAY "GFEDC":CFLASH0:CLS:COLOR,0:RESUME20
40020 IF ERR=53 THEN PRINT" ﾌｧｲﾙ ｶﾞ ｱﾘﾏｾﾝ----------":PLAY "GFEDC":CFLASH0:CLS:COLOR,0:RESUME20
41000 PRINT"  ｴﾗｰ ｶﾞﾊｯｾｲ-------------":PLAY "GFEDC":CFLASH0:CLS:COLOR,0:RESUME20
```

リスト3　MAIN-JOB 変更点

```
1 '---------MAIN-JOB---8ｶﾞﾂｺﾞｳｰ------ｽﾝﾎﾟｳｾﾝ ｼｮﾘ ｶｲｾﾞﾝ ﾌﾞﾌﾞﾝ---------
215 CLS
220 BEEP:E$="B":E1=1
1240 LOCATEINT((C4+C2)/16),INT((C5+C3)/(16*16)):PRINT SUN$:C=7
1241 '-------------------------------------------------------------
```

リスト4　(参考)

```
------------------ﾆｭｰ ﾍﾞｰｼｯｸ ﾀｲｵｳ ﾍﾝｺｳﾃﾝ---------------------
--- MAIN-JOB ----
440 LOCATE79,PT:PRINT" ":PT=INT(Y/8):LOCATE79,PT:PRINT"■"
-----ﾊﾟｰｽ----------------
2005 'COLOR6:LINE(49,3)-(79,24),"*",B  ⎫
2200 'COLOR5:LINE(0,15)-(46,24),"o",B  ⎭ REM化する


----ｿﾘｯﾄﾞ ﾊﾟｰｽ------
2830 'COLOR6:LINE(49,3)-(79,24),"*",B  ⎫
3050 'COLOR5:LINE(0,15)-(46,24),"o",B  ⎭ REM化する
```

ペンギン情報コーナー

WD-50

●NEW PRODUCT

新型ワードプロセッサ WD-2100, WD-50/55発売 シャープ

これまでのヨコ書き表示だけではなく，タテ書き表示も可能な本格ビジネス日本語ワードプロセッサ「ニュー書院WD-2100」が，シャープより750,000円(プリンタ別売) で発売された。

このWD-2100は，ビジネス用の日本語ワープロとして必要な作図機能，グラフ機能，抽出／分類機能，演算機能などに加えて，表計算型の簡易言語「書院カルク」が標準装備されているため，総合OA機器としての用途がさらに広がった。

WD-2100の主な特長は，

1. ただタテ書き印字ができるだけではなく，画面もタテ書きで表示。
2. 表示部は14インチのワイドで見やすいCRTディスプレイを採用。長時間のオペレーションでも疲れを感じない。
3. 文節変換だけではなく，まとめて入力した文章をあとで文節単位で変換できる一括選択方式を採用。
4. 通常の日本語の文章だけではなく，ルビ，分数，表を含む複雑な文書作成の際，ある特定の行だけの行間ピッチを変更し改行する特殊改行機能を採用。
5. 仕上がりの印字イメージを¼，⅙，⅛に縮小して画面に表示できるレイアウト確認機能を採用。
6. 2ページ分の印字内容を½に縮小し，1ページにまとめて印字することが可能。
7. 作表・集計もこなす簡易言語「書院カルク」を標準装備。
8. 多くの情報一覧の中から指定した内容を選び出したり，情報を並べ替えるソート機能を採用しているために，必要な情報の抽出，分類が簡単にできる。
9. ビジネス文書に欠かせないグラフ作成，作図機能も充実。
10. オプションにドットインパクト方式のプリンタWP-220DP，WD-220BP(各350,000円)，熱転写方式のWD-200TP (88,000円) を用意。

WD-2100

また，シャープではこのWD-2100に加えて，普及型パーソナルワープロ「ファミリー書院WD-50/55」(各69,800円)の発売も開始した。

このWD-50(JIS配例準拠キーボード)とWD-55(50音配列キーボード) は，美しい明朝体の24ドット印字，ハガキ印字が可能と，本格的機能を備えながらも7万円を切った価格で今回発売されたもので，約3万1千語の辞書を内蔵するなど，使いやすさを追求したワープロである。

WD-50/55の主な特長は，

1. 文字パターンは24ドット構成の明朝体を使用。また，官製ハガキへの印字も可能な新開発プリンタを採用。
2. 固有名詞 (姓，名前，地名) 約7千語を含む約3万1千語の辞書を内蔵。
3. 十二支，星座，お天気マークなどの120種類のイラスト記号の表示可能。
4. 案内状などの作成に，文章のまわりを線や記号で囲むことのできる枠組み機能を採用。
5. 縦横各1～24倍 (最大約8×8cm) まで文字や記号を拡大印字可能。
6. 32字までの常用句登録，10種類までの外字登録が可能。
7. ギリシャ文字やX^2，H_2Oなどの上つき下つき文字など，特殊記号345種類の使用が可能。
8. 電源を切っても約1300字(A4判1.3枚分)まで記憶できるメモリ保護機能付き。また カセットインタフェイス内蔵のため，カセットテープに文章を記録することもできる。
9. 小型，軽量 (329.5×300×51mm，2.6kg) のコンパクト設計で，カラーも黒と白の2色が用意されている。

〈問い合わせ先〉
シャープ㈱ ☎07435(3)5521

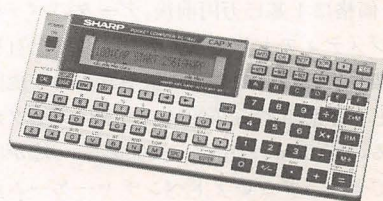
CAP-Xソフト搭載 ポケコンPC-1440新発売 シャープ

情報処理技術者試験の第1種，2種の出題科目であるCAP-Xソフトを搭載，さらにBASICを採用し，n進 (2，8，16進) での論理演算機能を充実させた多機能ポケコンPC-1440 (17,800円) が12月18日より新発売される。

このPC-1440は，CAP-X命令をアルファベットキーに割り当て，ワンタッチ入力を可能にしている。

PC-1440の主な特長は，

1. 情報処理技術者向けプログラミング言語CAP-Xソフト搭載。
2. アセンブラプログラミングに欠かせない2,8,16進数での論理演算(AND，OR，NOT，XOR) と補数計算が可能。
3. 16進データが入力しやすい独立16進キーを採用。
4. PC-1400シリーズのソフトが活用できるBASICを採用。
5. [CAL] キーを押せばワンタッチで関数電卓に切り替え可能。

〈問い合わせ先〉
シャープ㈱ ☎07435(3)5521

●INFORMATION

豪華商品が当たる
X1発売3周年記念キャンペーン実施中
シャープ

シャープでは，X1発売3周年を記念してX1turboIIを発売したが，それにあわせて11月16日から1月15日までの2カ月間，ユーザー向けキャンペーンをも全国で展開している。

このキャンペーンでは，全国のパソコンショップや販売店に置いてあるアンケートに答えると，その場で素敵なキャンペーンステッカーがもらえる特典があるほか，アンケートに付いているクイズに答えると，シャープビデオムービーやドットプリンタ，そのほか豪華商品が抽選で当たるなど，さまざまなイベントが用意されている。皆さんも一度，お近くの店頭をのぞいて見てはいかがでしょうか。

〈問い合わせ先〉
シャープ　☎06(628)1653

Again Watch

■1986・1
〔連載1周年記念〕
ニューイヤースペシャル

大特集Q&A：AからZまで

早いもので連載1周年記念を迎えた。何かしようと思い悩んだ結果，私が知る限りの範囲の中で，1986年度に予測される変化のうち皆さんが知りたそうなことをとりまとめて答えてみよう。Oh！MZが贈る第3の質問箱にまで発展するかは予断の許さないところだ。さあ，何でも聞いてくれ！ 私は知らないこと以外はすべて知っているのが自慢なのだ。

[A]PPLE社の新型パソコンが出るのはいつですか？

K.T　ズバリ86年1月である。32ビット機(MC68020) でハイスピード，しかもマッキントッシュより安い，ホームユース志向の機種ではないかという情報もある。

[B]ASICの日本語化は進みますか？

K.T　進む。たとえば命令語をこれまでの英語表記から日本語表記に置き換える方法もあるし，別の方法――たとえば，富士通では表示とプログラミングのときだけ日本語で，内部処理やセーブ時は元の英語版コマンドに戻る方法も，いくつかアプローチがなされているようだ。

[C]D-ROMの商品化概要を知りたい。

K.T　投入時期は春のマイコンショウあたり。発売メーカーは東芝，ソニー，日立，松下，パイオニアが確定。いずれもMSX2用で，価格は1基15万円前後。データのパッケージメディアとして考えられており，百科辞典や文献などの電子出版や写真入り地図などが予想される。600Mバイトのデータが入るので，FM-77AVだと6000画面の絵が収録できる。すごいアドベンチャーゲームができる。シャープではX1グループが検討しているが，膨大なデータを提供するメディアなので，メーカー独自機種では難しいかもしれない。ソフトの移殖のようにホイホイとはいかない。

[D]ECなどのミニコンはどうなるんでしょうか？

K.T　はっきり言って，パソコンとミニコンの機能差は縮まる一方。そこでミニコンメーカーは何か専門分野が必要となるが，それは通信制御と人工知能だと言われている。DECがVANと人工知能に全力投球しているのはそのためだが，もしそれもパソコンにとられたら……いちパソコンメーカーになるか，あるいは廃業しかない。この方向性は'86年度中に明らかになる。

[E]Cとの貿易摩擦はパソコン業界では？

K.T　ないとは言えない。すでに欧文タイプライタで日本メーカーがダンピングしているとして2～3割の輸入関税をかけている。複写機や油圧ショベルもすでにやり玉に上がっており，あと20種類の商品が候補に入っているそうだ。パソコンは恐らくその中のひとつであろう。

[F]AXとパソコンはつながりますか？

K.T　もちろん。'86年のエポックのひとつになるだろうが，パソコンにイメージスキャナを付けるだけで絵が入力でき，あとは通信機能を使って送受信するのはたやすい。印刷はプリンタだ。すでに数社が試作を終え，また超極秘情報だがNTTが商売として考えており，PC-9800はすでに確認済み。今MSX2でできるかどうかを某社に研究させているところだ。

[G]M-MAPって何ですか？

K.T　会社の中でコンピュータ同士をつなぐのはLAN（[L]参照）だが，工場専用のLANもあっていい。この工場用LANについて米国GM（ゼネラル・モータース）社が'83年からIBM，DECなどコンピュータ8社と決めた通信手段がMAPという。さき頃モトローラが通信用LSIの出荷を開始するなど準備中で，'87年春頃から使えるようになるそうだ。ホスト，パソコンのほかにNC制御装置，ロボット，計測機器までつなぐものだ。

[H]A（ホームオートメーション）はどういう形で実現しますか？

K.T　①セコムのやっているホームセキュリティ②パソコン通信のようなデータ通信系③テレビ中心のAVシステム系――の3系統があるが，どうやらバラバラに進みそうだ。商談はメーカーが住宅会社に売る形で進んでいるが，②だけはエンドユーザーにも比較的早く売れそう。

[I]BMの日本でのパソコン分野での攻勢は？

K.T　ゆっくりで，それでいて遊びがないと言える。ビジネス用は'85年に5540と5560を追加したことで着実だが，全体の売り上げはまだ全社の約3％を占めるにすぎない。家庭用はJXを便宜上続けている形だが，ニューマシンに切り替える話は私の耳には入っていない。ただし値下げはある。すでに定価の4割引きで売られている。

[J]OYSTICKやマウスの将来性は？

K.T　ジョイスティックはファミコン，MSXの発売以降，改良が進んでいる。今後は標準搭載する機種も含めて採用が増えよう。ただし，トラックボールという球形の装置もチラホラ出ているが，これは米軍の軍用機で使っているほど操作性がよく，ライバルとして浮上しよう。

マウスはビジネス用CAD/CAMを中心に完全に普及した。ライバルはライトペンだが，フラット形ディスプレイが出てくるとマウスが有力になる。

K T氏がAgain Watchの執筆から降りるという噂を聞きましたが……。
K.T　恐らく現実のものとなる。私は今年半ばで本業で地方に飛ばされそうなのだ。
L ANの市場はどうなりますか？
K.T　ワークステーションという名の高級パソコンを束ねてひとつの力にするための配線として，イメージが固まってきた。オフィスプロセッサの登場（O参照）もLANの実体が伴ってきた具体的な事象であり，'86年が真の意味でのスタートだ。これまでのような「特別な製品」からプリンタケーブルのように気軽に使うようになる。したがってメーカーはどこでもいい。
M SXの近況を教えてください。
K.T　主幹会社のアスキーによれば，'85年度末には累計出荷台数が100万台に達するそうだ。それは国内14社が競い続け，在庫がたまっても次から次へと新モデルを発売し続けた努力（？）の結果だ。まさに利益なき血まみれの戦いだったが，とにかくユーザー数も70万人はいそうで，やはり「共通性」の意味から見ると恐怖のパワーだ。
N TTのパソコン戦略が知りたい。
K.T　ビジネス用は「コムエクセル」という名で，PASOPIA1600を売っているが，大したことはない。
　狙い目はもちろん通信で，'86年10月に「NTT・PCコミュニケーションズ」という子会社がパソコン通信の交換センター業務をはじめる。電話のように，リアルタイムのパソコン通信仲介業を中心にするから，チャタリングなどの同種のサービスだけはアスキーネットなど他のパソコン通信サービス会社では手が出ない。結局，長いものに巻かれることになるのが'86年のパソコン通信市場と言える。
O ffice Computerはどうなるのですか？
K.T　実は中小企業のホストとしてのオフコンは'85年にすでに死んだ。'86年からは汎用コンピュータとパソコンの間の調整役としての役目を兼ね備えた発展的な商品である「オフィスプロセッサ」として再スタートを切る。パソコンのマネージャーと言えるだろう。もちろんネットワークとの通信サーバーとしての役目も果たす
P Cシリーズ（NEC）の独走は今年も続くのでしょうか。
K.T　何を言うか，パソコンを①MSXなど低価格機，②88など家庭用中・高級機，③98などビジネスパソコン――の3つに分けると，①は徹退したようなものだ。②もturbo IIやFM-77AVなどが出て，以前ほど強くはない。'85年の時点では②と③で強かったが'86年には独走は③だけになる。しかしパソコン市場でNECのトップがゆらぐのはしばらく先だろう。
Q Dの行く末はどうなるのでしょうか？
K.T　はっきり言って，パソコンのメモリがもはや64Kバイトを超えてしまった今では，パソコンの外部記憶装置としては苦しい立場だ。しかし世の中には64Kバイト以下ですむ仕事も多い。案外と電卓やワープロの文書ディスクとして普及するかも。
R obotとパソコンはこれから関係が深くなるかなあ？
K.T　もちろん！これも'86年の期待分野だ。すでにファミコンでは，ロボットを制御して遊べる。「プラレス三四郎」ではないが，パソコンにロボットをつないで戦わせる日も近いのではないか。「全日本ロボット格闘技選手権」なんてスゴイぞ。そのときに改めて「パソコンは格闘技だ」と言えるんじゃないのか。
S HARPで'86年に変わることは？
K.T　さあ。佐伯旭社長がそろそろ会長になるのではないかな。
T ele-Communicationはどの程度まで進みますか？
K.T　VとNを読んでもらえばわかるが，'86年は実感として進んでいくことがわかる年になる。ただしこれはビジネス分野の話で，家庭用パソコン通信は第1段階を迎えるのにもあと2年はかかりそう。
U NIXは普及するのでしょうか？
K.T　ほかにない以上，やはり普及してしまうでしょう。これは私の推測だが日本ではLANで接続したコンピュータ間（ホスト，オフコン，パソコン）で，共通利用できるソフトを作るための道具として使われるのではなかろうか。同じOSでもパソコン専用のMS-DOSとはそこが違う。
V AN（付加価値通信網）って何ですか。私たちとも関係あるんですか？
K.Tもちろん。定義は「異なるプロトコル（通信手段）間のコンピュータでも通信できるネットワーク」で，会社――銀行間など，まあコンピュータ間の広いネットワークと考えればいい。パソコン通信はもちろん含まれるだろうし，家庭にあるパソコンとホストをつなぐVANもPC-VAN（NEC）などいくつか出てきている。'85年はVANが実際に使われ出す年だ。ただし「できる」のと「使う」のは違う。家庭での普及はずいぶんと先の話。
W Pのブームはこれからもまだ続きますか？
K.T　もちろんです。ポータブルワープロは人気爆発で，'85年は'84年比4倍の80万台が売れたそうだが，'86年も'85年比5割増の120万台は当確だ。ただ，ひと言，売れ方の文具志向が強まっていたので，パソコンの需要層とは遊離していくようだ。つまり競合しなくなる。
X 1シリーズはこれからどう発展するのでしょうか？
K.T　その前になぜX1が後発なのにこれだけヒットしたかを説明しよう。性能の高さもさることながら，あのパソコンはビデオやテレビの担当者が家庭向きのパソコンを作ろうと開発した点にある。ほかはすべてコンピュータや電卓の担当者が作ったために，いまいち家庭用としてなじめなかったことは否めない。すでにテレビ，VTR，VHDビデオディスクがつながっているが，今後CDなどAV系の家電製品が次々にX1に接続されていくだろう。将来像はHAの中核機器になりそうだ。
Y AMAHA（日本楽器製造）のパソコン市場での狙いは何でしょうか？
K.T　MSX積極派の1社だが，将来的には音楽分野とニューメディア市場との融合を狙っており，そのキッカケとしては悪くなかったと見ていい。
　もうひとつ，LSIのビジネスも重要で，MSX2では東芝と並び主要LSIの供給メーカーとしてアスキーから指定されている。FM音源LSIも売れているし，今後，特化したカスタム市場での有力企業になりそうである。
Z 80などの8ビットマイコンはどうなりますか？
K.T　何が何でも16ビットではないのがマイコンのおもしろいところで，未だに4ビットマイコンは生産量で他をしのいでいる。MSXなどは一例だが，'86年はどんな用途に使うべきかを，利用者と生産者が確認しあう形になろう。
　以上。感想や別の質問をくれたまえ。Kのところで書いたように，私の命は残り少ない。(K.T)

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望のプレゼント番号をはがき右上のスペースにひとつ記入してお申し込みください。締め切りは１月15日の到着分までとします。なお，当選者の発表は３月号で行います。

愛読者プレゼント
新年特別拡張版

BASIC HOUSE ☎0286(33)1994

増設RAMボード KGB・128KBMZ

MZ-2500用 5名

特価9,800円

MZ-2500用の増設RAMでシャープ純正のMZ-1R26（35,000円）と完全コンパチです。

デジック ☎0899(24)0914

SUPER RAM BOARD MB-1000

X1/X1turbo用 2名

99,800円

大容量１Mバイトものメモリを持つ外部RAM。便利なユーティリティも付いている。

3

キャリー・ラボ ☎096(363)0047

JET-X1

X1/X1turbo用 5名

5D：35,800円

日本語ワードプロセッサのロングセラーJETシリーズの最新バージョン。X1用と，X1turbo用の2つのモードがある。

日本マイコン販売 ☎03(366)3274

テラ

X1turbo用 3名

5D：32,000円

パーソナルにビジネスにと威力を発揮する日本語ワードプロセッサ。16ビットからX1turboに移植されて登場。

5

BPS ☎045(421)7421

ザ・ファイヤークリスタル

X1/X1turbo用 3名

T：4,800円

大ヒット，ブラックオニキスの第２弾！より本格的なRPGとなって登場。

学習研究社 ☎03(720)1111

X1ターボ/X1シリーズ 活用プログラム特選集

システムハウス P&Pマイコン研究会編

10名

2,300円

先月号でご紹介した，BASICプログラムの学習書，基礎からじっくりと勉強したい人に最適。

7 Super mz 発売記念プレゼント

シャープ提供による豪華プレゼント第2弾！　今月はMZシリーズ用ソフトをドカーンと大放出。今では入手が困難な名作ソフトもあり今回が最後のチャンスとなるかも。

7-F (東海クリエイト) ☎03(456)4610

S・CALGO

MZ-700/1500用 100名

T：9,800円

CREATE LIBRARY
MZ-700 テープ 表計算・簡易言語 S・CALGO

7-A

System Program MZ-5Z002

MZ-700/1500用 10名

QD：10,000円

7-G (パーソナルメディア) ☎03(490)8841

グラフ8

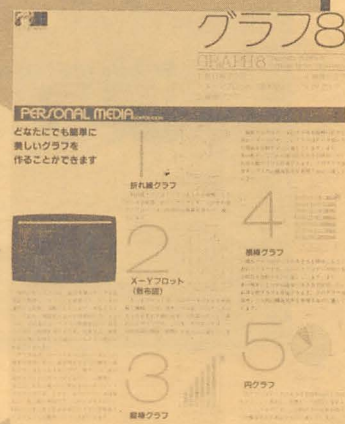
MZ-700/1500用 100名

T：12,000円

7-B

PASCAL Interpreter MZ-1Z004

MZ-2000/2200用 10名

T：12,000円

7-H

MZ-LOGO

MZ-2000用 100名

T：9,800円

7-I

MZ-LOGO

MZ-2200用 100名

T：9,800円

7-C

System Program MZ-1Z005

MZ-2000/2200用 10名

T：25,000円

7-D

Machine Language MZ-1Z006

MZ-2000/2200用 10名

T：7,000円

7-E

Floppy DOS MZ-2Z004

MZ-2000/2200用 10名

5D：50,000円

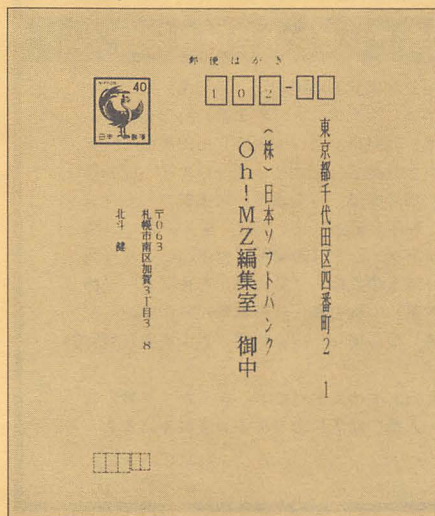
11月号愛読者プレゼント当選者発表

1スーパーカラーBASIC ⓐ（愛知県）高瀬博司　（静岡県）池田和弥　（新潟県）金田幸一　（岩手県）福永俊二　（秋田県）竹田忠雄　ⓑ（栃木県）岸孝弘　（埼玉県）西野寿泰　（大阪府）山田晃弘　（兵庫県）野田明男　（千葉県）大井英美　2オイルズ・ウエル　（神奈川県）永井克宜　（滋賀県）横井太道　（愛媛県）栗田智之　3マジックファクトリー（石川県）久保出紀史　（千葉県）桜井肇浩　（埼玉県）森田達也　4　ポケットブック　（静岡県）村田利弘　（岡山県）大西宣行　（青森県）清水頭武信　（愛知県）吉見裕史　（神奈川県）内瀬戸久司　（栃木県）小林由和　（熊本県）石坂康誠　（石川県）小林孝　（兵庫県）成田博一　（東京都）秋山賢史（敬称略）ほか40名の方。

以上の方々が当選されました。おめでとうございました。なお，賞品は順次発送いたしますが，入荷の状況によって多少遅れる場合もございますのでご了承ください。

ブラザーから発売されたこのプリンタは，低価格(本体定価128,000円)ながら，24ピンのドットインパクト方式で漢字を印字でき，また，その名のとおり型式の定まった用紙への印字を簡単に行えることや，はがきへの印字を簡単にこなしてしまうといった特徴もあります。これらについて実際に使用してみての操作性や印字例についての報告をします。

はがき印字も可能

このプリンタは，従来の漢字プリンタと同じ機能の他に，セールスポイントであるいくつかの特長を備えています。まずはその中のひとつである「はがき印字モード」について紹介することにします。

オンライン（はがき印字）スイッチを押しながら電源をONにすると，このモードを設定できます。このモードは官製はがき(私製のものでも大きさを同じにすれば可）に住所，氏名を簡単に印字できるようにするものです。普通のプリンタでは，はがきの位置決めやパソコンから送るデータのフォーマットが大変ですが，このモードではそれらをプリンタが自動的に行ってくれます。

印字するデータはワープロソフトや顧客管理ソフト，あるいはBASIC(2500, turboは付属のもの，X1はＮＥＷ BASIC＋KANJI. Uty)で作成します。1件のデータ数は6行までで，1行目を郵便番号(数字のみ有効)，2～6行目を住所・氏名データとして漢字で16文字まで使えます。あとははがきを枠に入れスイッチを押すと，自動的に印字位置にセットされるので，データを送ると所定の場所に印字されます。

「割付名人」は割付けの名人

もうひとつの特長は，書式の決まった用紙への印字を行う「書式印字モード」です。改頁（書式印字）スイッチを押しながら電源をONにするとこのモードが設定できます。

このモードは，見積書，注文書といった定型用紙への印字を容易にするモードです。すなわち，各印字項目（氏名，年月日，製品名，金額etc.）の印字開始位置がプリンタに記憶され，パソコン側からはデータのフォーマットに煩わされることなく，項目順にデータを送るだけで決まった様式の印刷を何枚も行うことができます。

また，印字開始位置の設定ですが，これにはパソコンから制御コードを送る方法(プログラム例がマニュアルに載っています）と，オプションのフォーマットキーボード（FK-20）を使う方法があります。前者の方法は多少面倒なので，ここでは後者のキーボードについて説明します。

このキーボードは書式の設定を簡単で確実に行うためのもので，登録した書式の記憶，呼び出し，修正などができます。設定

はがきの印字

サンプル1

```
100 '***************************************************************
110 '*「はがき印字のサンプル」                                      *
120 '*                                                             *
130 '* このようなフォーマットで印字すると、はがきに簡単に住所が印字できます。*
140 '*                                                             *
150 '*        DIP SW-No.4〔OFF〕の時は縦書きになります。           *
160 '*                                                             *
170 '*        DIP SW-No.4〔ON〕 の時は横書きになります。           *
180 '*                                                             *
190 '***************************************************************
200 '《差出人住所》
210 LPRINT"〒０６３"
220 LPRINT"札幌市南区加賀３丁目３－８        "
230 LPRINT"                                  "
240 LPRINT"北斗　健                          "
250 LPRINT"                                  "
260 LPRINT"                                  "
270 '《宛先人住所》
280 LPRINT"〒１０２"
290 LPRINT"東京都千代田区四番町２－１        "'倍角印字
300 LPRINT"                                  "'倍角印字
310 LPRINT"（株）日本ソフトバンク            "'倍角印字
320 LPRINT"　Ｏｈ！ＭＺ編集室　御中          "'４倍角印字
330 LPRINT"                                  "'４倍角印字
```

の方法は，書式を設定したい用紙をプリンタに入れて，キーボードのカーソルキーを押し，印字ヘッドを移動させながら各項目の印字開始位置に合わせて設定ボタンを押していくだけです。

書式を設定したあとは，パソコンから印字データを送るだけで印字できます。はがき印字と同様に，BASICやワープロで作成します。

総合評価

最初に，このプリンタについて好感を持った点を挙げると，多機能の割にはコンパクト（352×234×78mm）であること，また同じく24ドットの漢字ドットインパクトプリンタとしては安価であることなどです。機能としては，先に述べたようにメジャーなプリンタとコンパチなため使えるソフトが多いこと，はがき印字，書式印字といった便利なモードがあることが良い点です。

逆に改善されるべき点としては，印字音があります。このプリンタ自体は比較的音の小さい方ですが，狭い部屋で使用したためかなり気になりました。これは現在のドットインパクト式プリンタの宿命でしょう。また，印字速度が遅くて高速モード（最高約30行/分）でも，大量の印字にはイライラしてしまいます。

もうひとつは，書式印字モードは非常に便利なのですが，それをより活かすために用紙のセットをもっとしっかりできるようにしてもらいたいと思います。本体付属のペーパーガイドは支えが弱く，すぐにずれてしまいます。せっかく細かい書式が設定できるのに用紙の位置が不安定では何にもなりません。オプションのオートカットシートフィーダーを使えば良いのかもしれませんが，本体でももう少し配慮が必要です。

最近は，いろいろなプリンタが安価で高性能になって発売されています。規格の統一が望まれながらも，なかなか完全にはできていないようです。メジャーなメーカーのものに合わせていくといった，消極的な意味では多少は統一されつつある気もしますが……。先ほども触れたように，ソフト側で多くのプリンタをサポートしていくという方向も当然ありますが，せっかく良いプリンタが出ても自分のパソコンで使えなければ困ってしまいます。これからはプリンタの規格も統一されていってほしいものです。幸いにして，私が使用しているX1 turboやX1（FとNEW BASICに付属），2500にはこうしたプリンタの状況を踏まえて，プリンタ設定が可能なユーティリティがあるので困ることもなく新しいプリンタの発売は，周辺機器の選択の幅が拡がることであり歓迎できます。

X1/turbo，2500では，このプリンタを含めたNEC系のプリンタも使用できますので，プリンタが欲しいと思っている方はそのこともお忘れなく。　（浅野　恵造）

M-1024P（本体）	128,000円
FK-20（キーボード）	29,800円
PF-50(ピンフィードユニット)	5,000円
第2水準ROMボード	20,000円
オートカットシートフィーダー	20,000円 （1月末発売予定）
ブラザー販売㈱	☎052(263)5811

定型書式印字

御見積書　No. HL-6013

'61ハレー観光ツアー御一行　様　　2060年　3月　6日

受渡期日　2060年　4月　16日

受渡場所　ルナステーション弊社専用ポート

取引方法　連邦法第150条第2項準拠

有効期限　2060年4月30日

合計金額　ε$207,000

普及型6人乗りシャトルMZturbo	1	128,000	128000
Z8エンジン用燃料(U-300)	12	4,800	57600
生命維持パック(LS511)	5	2,500	12500
乗員用ハンディパソコン(CZ-888C)	5	980	4900
音声合成・認識ユニット(CZ-888V)	5	680	3400
通訳ソフト「エスペラント」(JP5/V.2.2)	5	120	600

合　計　207000

サンプル2

```
100 '**************************************************************
110 '*                                                            *
120 '*「書式印字のサンプル（見積書）」                            *
130 '*                                                            *
140 '*  このようなフォーマットで見積書に簡単に印字ができます。    *
150 '*                                                            *
160 '**************************************************************
170 LPRINT"101"'---------------------------------見積書様式
180 LPRINT"HL-6013"'-----------------------------見積書No.
190 LPRINT"  2060     3     6"'-----------------発行年月日
200 LPRINT"'61ハレー観光ツアー御一行"'-------------お得意様名
210 LPRINT"2060     2   29        004"'------------照会年月日No.
220 LPRINT"２０６０    ４    １６"'-------------受渡期日
230 LPRINT"ルナステーション弊社専用ポート"'---------受渡場所
240 LPRINT"連邦法第１５０条第２項準拠"'-------------取引方法
250 LPRINT"２０６０年４月３０日"'-------------------有効期限
260 LPRINT"ε＄２０７，０００"'---------------------合計金額
270 LPRINT"普及型６人乗りシャトルＭＺturbo"'--------適要１
280 LPRINT"     1   128,000        １２８０００"'数量／単価／金額１
290 LPRINT"Ｚ８エンジン用燃料(U-300)"'---------------適要２
300 LPRINT"    １２    4,800          ５７６００"'数量／単価／金額２
310 LPRINT"生命維持パック(LS511)"'-------------------適要３
320 LPRINT"     ５     2,500          １２５００"'数量／単価／金額３
330 LPRINT"乗員用ハンディパソコン(CZ-888C)"'--------適要４
340 LPRINT"     ５       980            ４９００"'数量／単価／金額４
350 LPRINT"音声合成・認識ユニット(CZ-888V)"'--------適要５
360 LPRINT"     ５       680            ３４００"'数量／単価／金額５
370 LPRINT"通訳ソフト「エスペラント」(JP5/V.2.2)"'--適要６
380 LPRINT"     ５       120              ６００"'数量／単価／金額６
390 LPRINT""'------------------------------------適要７
400 LPRINT""'------------------------------------数量／単価／金額７
410 LPRINT""'------------------------------------適要８
420 LPRINT""'------------------------------------数量／単価／金額８
430 LPRINT"        ２０７０００"'--------------------合計
```

SYSTEM HARDWARE REPORT

MZ-2500, X1/X1 turbo 話題の拡張メモリボード

最近，シャープの純正品以外のMZ-2500用の拡張メモリボード，拡張ビデオRAMボード，大容量外部メモリボード，X1/X1 turbo用大容量外部メモリボードが相次いで発売されました。純正品と互換性があり，しかも低価格なこれらのRAMボードは，ユーザーとしても見逃せないところですね。今回はその試用レポートをお届けします。

MZ-2500用メモリボード	
増設RAMボード（BASIC HOUSE）	
KGB-128KMZ	12,800円
（現在5周年記念特別価格 9,800円）	
増設RAMカード（テレシステムズ）	
RM-25A-2	12,100円
増設ビデオRAMカード（テレシステムズ）	
RM-25A-1	13,100円
（RM-25A-1, 2のセットRM-25は24,800円）	
640KB RAM DISK（テレシステムズ）	
RM-25E	49,800円
X1/X1 turbo用メモリボード	
512KB RAM DISK（テレシステムズ）	
RM-X1E	49,800円
SUPER RAM BOARD（デジック）	
MB-1000	99,800円
〈問い合わせ先〉	
BASIC HOUSE	☎0286(33)1994
(有)テレシステムズ	☎06(631)0925
(株)デジック	☎0899(24)0914

MZ-2500用各種メモリボード

BASIC HOUSEから発売された増設RAMボード(KGB-128KMZ)，テレシステムズから発売の増設RAMカード(RM-25A-2)，増設ビデオRAMカード(RM-25A-1)，640KB RAM DISK（RM-25E）を紹介しましょう。

KGB-128KMZとRM-25A-2は，シャープ純正の拡張RAMボード(MZ-1R26）とは少し形が違うだけで，機能的にはフルコンパチの128Kバイトメモリボードです。価格は，KGB-128KMZが12,800円，RM-25A-2が12,100円ですが，KGB-128KMZはBASIC HOUSEの5周年記念特別価格として現在 9,800円で手に入ります。また，RM-25A-2のほうは後述のRM-25A-1とセットになったRM-25Aが24,800円です。いずれにしても，純正品(35,000円)の⅓ぐらいの価格です。実際256KビットのDRAMが4つささっているだけなのに，純正品はどうしてあんなに高いんでしょうね。

MZ-2500の本体内スロットに実装してみると，いきなりBASICのフリーエリアが約169Kバイトになりました。マウスや文節変換を使っても153Kバイトあるのですから，安心して文節変換を使った大きなプログラムも組めるようになります。

RM-25A-1は，シャープ純正の拡張ビデオRAMボード(MZ-1R27)とフルコンパチの64KバイトV-RAMボードです。純正品が20,000円なのに対してこちらは13,100円になっています。

増設ビデオRAMカードを装着すると，一気にグラフィックV-RAMが128Kバイトに増え，640×400ドットの16色表示や320×200ドットの256色表示が2画面など，16ビット機にも勝るグラフィック機能が実現されます。また，MZ-2500のBASICではグラフィックV-RAMをメモリディスク（メモリの一部を外部記憶装置のように使う）やプリンタスプーラ（バッファ）として使えるようになっているので，派手なグラフィックを必要としない場合でも付けておいて損はないと思います。プリンタスプーラとは，一般にプリンタの動作中のパソコン本体もそれにお付き合いするために，動作終了までほかの仕事は何もできないという状況を打開するため，メモリの一部にプリンタの印字データを送っておき，それをプリンタへ送りながら並列してほかの仕事を行える機能です。この機能を使えば長いリストをとっている最中でもBASICのプログラムを作成したり，走らせたりすることができるのです。

グラフィックV-RAMは128Kバイトまでのメモリディスクとしても使えましたが，テレシステムズの大容量メモリボードRM-25E（49,800円）はディスク1枚分の640Kバイトの容量を持つメモリディスク専用ボードです。こちらは純正品が出ていませんが，BASIC, P-CP/Mでサポートされていますので，安心して使うことができます。

これを本体に装着するにはシャープ製の拡張ユニットMZ-1U09(9,000円)が必要です。このユニットの本体側のコネクタはドライブの隣りの奥まったところにあり，慣れない人には取付けが非常に難しいと思われます。この程度の価格で出せるのですから，ぜひとも標準装備にしてほしかったと思います。

では，実際に使ってみましょう。RAMディスクを使うための命令がBASIC上に用意されています。キーワードは“EMM:”。X1/X1 turboと同じですね。フロッピーディスクに対する命令はすべて“FD:”の代わりに“EMM:”を使うだけでRAMディスクに対する命令となるのです。RAMディスクを使うには，まずINIT“EMM:”を実行しRAMディスクを初期化します。そして，たとえばRAMディスク上にランダムアクセスファイルをオープンしたければ

OPEN“R”，#1,“EMM:ファイル名”

とすればよいのです。同様にフロッピーの内容をすべてRAMディスクに移すには本体付属のユーティリティ“フォーマット&コピー”を使えばよく，CHDIR“EMM:”でカレントディレクトリがRAMディスクに移るといったようにいたって簡単に使えます。しかも，相手はRAMですからフロッピーディスクに対するときとは比べものにならないほど高速に（約10倍）書き込み／読み出しができるので，プログラムやデータのセーブ／ロード時のイライラは完全に解消されるのです。ファイル処理をよく行う人にはかけがえのないものといえるでしょう。

不満といえば，MZ-2500に付属のコピーユーティリティがいささか使いにくく，ディスク１枚分をＥＭＭにフルコピーするのに１分ほどかかってしまいます。EMMを有効に活用するためにも，高速のコピーユーティリティなどが欲しかったところです。

このRAMディスクは，P-CP/M上でもドライブ名“F:”としてすぐ使えるようになっています。CP/MなどのDOSでは，BASICに比べてディスクアクセス回数がかなり多いので，このRAMディスクを使用すれば，フロッピーディスクの摩耗防止だけでなく，アクセス時間待ちのイライラ解消にもなります。X1とCP/Mでプログラム開発などをしてる人がＥＭＭを有効活用しているのを見ても，その威力のほどがうかがわれます。

このように，RAMディスクはフロッピーディスクの代わりどころかそれ以上の可能性を秘めたボードといえます。あとは，優秀なユーティリティソフトと電池などによるバックアップがあれば，完全にフロッピーディスクと置き替わることができるのではないでしょうか。

なお，上記の各種メモリボードは MZ-80B/2000モードでは使用できません。

（吉田　幸一）

X1/X1 turbo用外部RAMボード

すでに広告や本誌85年10月号の情報コーナーでご存じでしょうが SUPER RAM BOARD (MB-1000) がデジックから99,800円で発売されています。容量は１Ｍバイト＝1024Ｋバイトと，EMM（シャープ製320KB外部メモリCZ-8EM）の３枚分＋64Ｋバイトとなっています。機能はＥＭＭと完全にコンパチブルです。以下に付属のユーティリティソフトなどとともに見てみましょう。

MB-1000はCZ-8EM３枚分とコンパチブルです。CZ-8EMにはディップスイッチが付いており，EMM0:～EMM64:まで(BASICでのサポートはEMM0:～EMM9:)を選択できますが，MB-1000では２とおりの設定だけしかありません。すなわち，EMM0:～EMM2:に割り当てるか，EMM4:～EMM6:に割り当てるかです。そのほかの細かい設定はできませんが，実際に使ううえでの不都合などはないでしょう。ハードの点でほかに書くべきことは，余っている64Ｋバイト分へのアクセス方法ですが，EMM0:（もしくはEMM4:)からは，なんと1Mバイトをそのまま使えるようになっています。

では付属のユーティリティについて紹介しましょう。付属のユーティリティは４種類，７本あります(図)。

まずは“MBTEST”で，これはMB-1000の動作テスト用ユーティリティです。１Ｍバイト全体に対して，書き込み，読み出し，比較を行います。

次に“MBTEST”ですが，これはEMMとほかのデバイスの間でデータ転送を行うものです。各デバイスと方向は

① フロッピーディスク ↔EMM
② グラフィックメモリ ↔EMM
③ EMM ↔メインメモリ
④ EMM ↔EMM

が選べますが，残念ながら②～④はX1 turboでしか動きません。また，方向やデバイスの指定はPOKE命令で指定しなければならず，エラーが発生した場合のエラーコードもPEEK命令で読み出さなければなりません。マニュアルではその方法をちゃんと説明してあるのですが，やはりBASICによるサンプルプログラムぐらいあってもよかったのではないでしょうか。ユーティリティとしてユーザーが自作したプログラムとリンクする場合には特に問題があるわけではありませんが，少々不親切な気がします。

次に“LPTUTY”ですが，これはいわゆるプリンタバッファ機能を実現するソフトウェアです。BASICを書き換えて，LLIST，LPRINT，HCOPYの3つのコマンドについてプリンタバッファ機能をもたせるもので，これによってプリントアウトの間CPUが待たされることがなくなり，作業効率が上がります。ただしこれはCTCが必要なためturbo BASIC専用です。しかし，これはマニュアルには書かれていません。誤解を避けるためにぜひとも明記しておいてほしいところです。

バッファにはEMM0:～EMM9:,V-RAMの裏（バンク１）を指定することができます。バッファ機能を無効にすることもCALL命令で簡単にできますので，プリントアウトが多い作業には有効でしょう。

最後に“SEARCH & SORT”ですが，これもturbo用です。機能はVDIMで宣言した配列に対して高速検索，高速並べ換え（ソートをするもので，これにはBASICで書かれたコントロールプログラム例が付いています。使い方は少々複雑ですが，実数データのソートも可能ですから，使いこなすことができればなかなか強力なツールになるでしょう。

以上，このようなボードにユーティリティが付属していることはたいへん評価に値しますが，細かいことを言えば，特にX1ユーザーには少々もの足りない気がします。しかし，CZ-8EMが1枚88,000円であることを考えると，3枚分が99,800円と40%以下の値段で出てきたのは嬉しい限りですね。

次に紹介するのは，本誌85年11月号の情報コーナーでも紹介されたテレシステムズのX1用512KB RAM DISK(RM-X1E)です。これはEMM＋192Kバイトの機能と，PROM (BASIC ROMに対応)ボードの機能を１枚のカードに納めたもので，価格は49,800円です。EMMのアドレスはディップスイッチによりEMM0:～EMM15:まで選択でき，またPROMのアドレスも0E00H～0E03H だけではなく，指定された部分のパターンカット，接続で0E04H～0E07Hに変更することも可能です。PROMは付属しておらず，自分でPROMライタを使い書き込むなどして使うためのものです。

EMM部へのアクセスは00000H～7FFFFHまでのアドレスを指定することが可能で，そのようにすると512Kバイトをすべて使うことができます。

やはり気にかかるのは容量が中途半端な点で，256KビットRAMを使っているためなのでしょうが，どうも192Kバイトが無駄になってしまう気がします。また，マニュアルには，「IPLスイッチを押したりboot命令を実行すると」内容が消えると書いてありますが，そんなことはないはずです。

以上，２つのボードについて書きましたが，まさにDRAMが安くなっているという状況を表しています。この調子でいけばひょっとして，フロッピーディスクは単なるバックアップ用のデバイスにすぎなくなるのかもしれません。（高野　庸一）

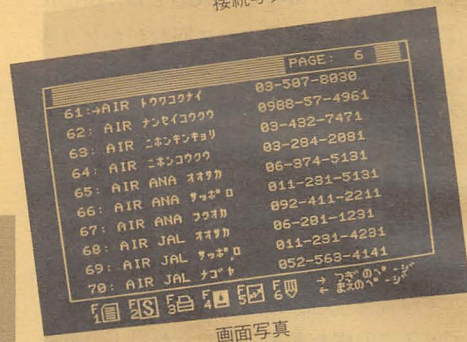
MB-1000に同梱のユーティリティ

```
Bin*    "1:MBTEST          .  "
Bin*    "1:MBTRANS         .  "
Bin*    "1:LPTUTY(E600)    .  "
Bin*    "1:LPTUTY(D600)    .  "
Bin*    "1:LPTUTY(C600)    .  "
Bin*    "1:SEARCH & SORT.     "
Bas     "1:S&S TEST`       .  "
```

MZ-1500用POKET DB

ポケットデータベースを街で使おう

Koichi Tetsuji
挙市 哲司

接続写真

画面写真

シャープからポケットDBインタフェイス(MZ-1E25)が発売されました。これはシャープの電子メモ，愛称ポケットDB(データベース)とMZ-1500の間で住所録のデータをやりとりするためのインタフェイスです。価格は9,980円で，これはパソコンのデータ入力方式に新しい風を吹き込むものとなりそうです。

電子メモとは？

電子メモとは，ひと言で言うと，電卓にアドレス帳の付いたようなものでしょうか。

もちろん，ただのアドレス帳ではなくて，サーチ機能や自動ソート（名前を五十音，アルファベッド，数字，記号の順に並べる）機能も付いていて，しかもパソコンと通信ができるという優れ物です。

容量は2KB(ユーザーエリア1856バイト)のものと4KB(同3904バイト）のものがあり，名前4文字，電話番号12桁を入れた場合，前者で150人分，後者で320人分のデータが入ります。名前の部分には他の文字データ（住所など）が80文字まで入りますが，これを入れた場合はもっと人数が少なくなります。

電子メモの価格
PA-300(2KB)13,800：以下(　)内は実装メモリ
PA-300D(2KB)19,800円：本皮手帳型ケース付，ゴールド仕様
PA-301(4KB)17,800円：今回紹介のもの
PA-302(2KB)15,800円：本皮手帳型ケース付
CS1000D(4KB)60,000円：音声時計内蔵，本皮手帳型ケース付

使える文字はカタカナ，アルファベット(大文字)，数字，そして若干の記号（－，〒，／，：，＊，スペース，「 」)です。カナの入力はローマ字カナ変換方式で行います。

機種によっては電卓機能の他に，ポケコン機能の付いたものがありますが，今のところ通信できるのはTELモードと呼ばれるアドレス帳のデータに限られています。

MZ-1500との接続方法

ポケットDBインタフェイスはMZ-1500のデータレコーダ端子に接続します。そのため，MZ-1500内部のジョイントコネクタを取り外す必要があります（データレコーダを使用し，すでにジョイントコネクタを外してある場合はこの作業は不要です)。

ジョイントコネクタを取り外したら，ポケットDBインタフェイスのMZ-1500側のコネクタを，MZ-1500後面のデータレコーダ端子に，そして電子メモ側のコネクタを，電子メモの接続端子に接続します。

接続は電源を切ってから，慎重に行ってください。また，接続中にMZ-1500のカセットに対する命令を実行すると電子メモのデータを破壊する恐れがあります。

付属ソフトウェア・アドレスノート

ポケットDBインタフェイスにはアプリケーションソフト“アドレスノート”が付属しています。電子メモとのデータ通信はこのソフトを使って行うためのものです。アドレスノートの主な機能を挙げておきます。

○QDへのデータのSAVE
○QDからのデータのLOAD
○個人データの入力
　個人データ255人分入りますが，電子メモの容量を越えては転送できません。
○データの一覧表表示
　電子メモと同じく，五十音，アルファベット(大文字)，数字，記号の順にソートします。
○個人データの表示，変更，削除
○サーチ機能
○プリンタへの印字
　個人データをダイレクトメール用ラベル形式に印字することも可能です。
○電子メモとのデータ通信
　MZ-1500から電子メモへ通信する場合，電子メモ上のデータは消されて，MZのデータが新しく入ります。このとき，転送するデータを指定できます。また，サーチ機能と組み合わせて，ある文字を含む名前のデータだけ転送することもできます。また，電子メモからMZ-1500に転送する場合は，メモリ上のデータに電子メモのデータが追加されます。通信に要

する時間は，最大約1分です。

アドレスノートでは，操作の指定はすべてファンクションキーで行い，どのファンクションキーにどの操作が対応するかも常に画面に表示されますから，マニュアルがなくても簡単に使えるようになっています。また，個人データやサーチするデータの入力には，カナ入力とローマ字入力を切り替えて使うことができます。

アドレスノートが入っているQDのB面にはサンプルの住所データと，ユーティリティソフトが入っています。これはMZ-1500本体付属の住所録ソフトのデータを，アドレスノートで使えるように変換するためのプログラムです。

街に出よう！

電子メモとMZ-1500をつないで，データのやり取りができるようになると，そこには様々な使い方が生まれてきます。データは何も住所，氏名，電話番号に限る必要はまったくないのです。さて，あなたなら何を入れてどう使いますか？

たとえば，次のような使い方はどうでしょうか。

メモ帳代わりに

あなたが，街のポスターあるいは電車の中吊り広告でマイコンショウの開催を知ったとします。いつ？　どこで？　普通なら，手帳を出してメモするところですが，あなたは違います。さっとポケットDBを取り出し，キーをたたきます。まわりの人が，おや？　とあなたを見ます。鉛筆で書くのと違って，間違っても消しゴムでゴシゴシやる必要はありません。チョンとキーを押し，カーソルを戻して訂正。実にスマートです。

おもしろそうな本や映画の題名，バーゲン情報等々，街には情報がいっぱいです。

あんなこと，こんなこと，すべてメモしておいて，部屋に帰ってMZにつなぎ，即プリントアウト。きれいな文字が残ります。

タウン情報

ぴあ（関東）やLマガ（関西）片手に歩くなんて，かっこ悪い。ポケットDBならMZに入れたタウン情報のうち，必要なものだけをサーチ機能で取り出して転送。今日は新宿，明日は難波，DB片手に行ってらっしゃーい。

時刻表

といって，街に出たのはいいけれど，はしゃぎすぎて，あれ，終電何時だっけ？　こんなことのないように，ちゃんとメモっておきましょう。

スケジュール表

ソート機能を利用して，便利なスケジュール表ができます。

MZ-1500の側で月／日／時／分という名前で各スケジュールを登録します。この場合，数字はすべて2桁にします。1月1日なら01／01です。こうすれば，ソート機能であとから追加しても自動的に並べ替えられます。そして必要な（この先1週間なら1週間分）データだけをポケットDBに転送します。

日記にもなる！

何かあったら，すぐポケットDBに書き込みましょう。帰ったら，すぐにMZに転送。日記専用のQDを作っておけば，追加された分だけ書き変わりますから，とても便利です。しかし，日付を2桁で入れることをお忘れなく。

パソコンの外部入力装置

ポケットDBをパソコンの外部入力装置として使いましょう。しかし，現在その入力データは住所録以外のデータとしては使えません。何らかのコンバートプログラムが必要です。コンバートプログラムがあるものとして，以下の記事は読んでください。

プログラムの入力

通勤，通学の電車の中で，あなたは何で時間をつぶしていますか？　本を読む人，貴重な睡眠時間にあてる人，様々ですが，この時間をプログラムの入力に使うことができるとしたらどうでしょう。通勤，通学の時間は短いようで，毎日積み重なれば長いものです。打ち込みたいプログラムがあるんだけど，忙しくてそんな時間がない。そういうときに，このポケットDBで少しずつ打ち込んでいくことができれば素晴しいですね。

ワープロに使う！

電車の中でふと浮かんだ名文句をチョンチョンチョンと打っておき，あとからワープロで漢字に変換する。俳句や和歌，詩が思い浮かんだらすぐに打ち込む。アイデアがすぐ文字になるなんて，素敵ですね。あらゆる場所があなたの書斎になります。

今はポータブルワープロが全盛の時代ですが，気軽に持って歩くには，いまいち大きすぎますね。その点，このポケットDBならシャツのポケットにもすっぽり。2KBの容量を多いと見るか，少ないと見るかは，あなたの文才次第です。

現在，MZ-1500用のワープロ，ユーカラjjの外部入力装置として使えるよう，データコンバータを検討中です。機会があれば，発表したいと思います。

通信新時代に向けて

パソコンと通信可能なポケットDBの出現は，新しいメディアの先駆けとなるかもしれません。近い将来には，街のあちこちで端末にポケットDBにつないでは，データをやりとりする風景が，当たり前のことになっているでしょう。

そうなれば，あなたが毎月（毎週）買っている雑誌の情報なんかは，簡単に手に入れることができるはずです。

今日はポケットDBをメモ代わりにして，せっせと情報を書き込むことをお薦めしましたが，こうなると書き移す手間がなくなって，街のあちこちの端末につないでは，あっちの映画情報，こっちのアルバイト情報と，次から次へと新しい情報が得られるはずです。

そして，もっと進歩してくれて，サイズはそのままでもメモリが大幅にアップして，さらに街の公衆電話から，いとも簡単に自分の部屋のパソコンなんかに情報を送り込めるようになってくれれば，きっとライフスタイルから変わってしまうかもしれませんね。

たとえば，毎日読んでいる新聞なんかは，写真まで見ることができないにしても，その日の情報が，次から次へと24時間，街の端末から提供されるとなればいったいどうなってしまうのでしょう。テレビやラジオという，現在の電波メディアの在り方すら変わってしまうかもしれない大事件にもなりかねませんね。

それにしても，数年前にウォークマンが流行したときと同じように，今度はヘッドホンの代わりに，街のあちこちでポケットDBを手にしている姿を，想像するだけでも楽しいことですね。

FILES Oh!MZ

このインデックスは，タイトル，注記――著者名，誌名，月号，ページから構成されています。今月はMZ-2500関係の紹介記事が多いようです。

I/O　工学社
ASCII　アスキー
テクノポリス　徳間書店
Pio　工学社
POPCOM　小学館
マイコン　電波新聞社
マイコン　BASIC Magazine　電波新聞社
マイコンライフ　学研
LOGIN　アスキー

『パソコンでVHDを楽しむ本』というたいへん軽いタイトルであるが，中身はきわめてテクニカルな内容の本である。VHDおよびパソコン-VHDインタフェイス関係の技術資料をひとつにまとめたものといえよう。現在，VHDについてこれほど詳しく書かれた本はほかにない。内容は大きく，①VHDの仕組み，②VHD言語，③VHD活用法，に分けることができる。①はVHDシステムの原理を中心にした技術解説である。②はVHD言語の仕様や使い方について詳しく解説されている。現在VHD言語によるプログラム開発環境はユーザーには提供されていないので，これは将来のためのマニュアルと見るべきであろう。③はユーザーがもっとも関心のあるところである。Oh!MZ 85年10月号で紹介したVHDインタフェイスユニットVO-20PCを中心に，これの使い方や一般のVHDディスクを使ってのサンプルプログラム(X1用)が紹介されている。全体に専門的な内容であり，どちらかといえば「VHDシステムのすべてがわかる本」ということができる。(N.N.)

パソコンでVHDを楽しむ本
日本ビクター 平田渥美著
工学社　B5判　216ページ　2,500円　☎03(375)5784

一般

▶あれば便利な小道具大道具
一気一気で整理整頓をしてしまおう！ そのためのツールを君に教えちゃおう。――冨山学，マイコンライフ，12月号，18―24pp.

▶中古市場に注目!! 知らないと損する中古品入手のコツ
中古パソコンなんて，と言う前に知っておけば便利な中古品購入術について。――藍澤公平，マイコンライフ，12月号，50―55pp.

▶この許せない話
もっとメーカーはユーザーサイドに立ってくれ，というお話。皆さん怒りましょう。――ピーターパン富野，マイコンライフ，12月号，60―61pp.

▶中古パソコン学のススメ　少ない予算をフル活用する
ワンランクアップを図ることができる中古パソコン購入学。――安藤欣太郎，マイコン，12月号，311―314pp.

▶購入＆買い換え大作戦
必ず成功する，というパソコンの購入，買い換えのテキストです。――編集部，テクノポリス，12月号，76―82pp.

MZ80K/C/1200/700/1500

MZ-80K/C/1200

▶UFO GAME
UFOを操って，Pを取るゲーム。――しんちゃん先輩，Pio，12月号，124p.

▶いないいないバア～
地面の色をかえていくというワンキー操作のゲームです。――てつろうくん，マイコン BASIC Magazine，12月号，87p.

▶ザ・カラテ2
格闘技ゲームです。燃えてください。――宇都宮伸吾，Pio，12月号，82―91pp.

▶TENSION
あなたはガンマン。家の中に隠れている悪党共をやっつけてください。――ELECT RINSE，Pio，12月号，124 -125pp.

MZ-700

▶ゼラートル
ゼラートルという怪物を倒すロールプレイ・ゲームです。――平井真二，Pio，12月号，33―35pp.

MZ-1500

▶トリサン mk1'sアルバイト
アルバイトを探すトリサンmk1の物語です。――斎藤将之，マイコン BASIC Magazine，12月号，92―93pp.

MZ-700/1500

▶OTTA & TOTTA
トッタクンに帽子を取らせるゲームです。――小幡利顕，マイコン BASIC Magazine，12月号，90―91pp.

▶バスケットボール・ゲーム
コンピュータと一対一のバスケットゲームです。――吉田拓，マイコン BASIC Magazine，12月号，88―89pp.

MZ-80B/2000/2200/2500

MZ-80B/2000/2200

▶ダブルボール
2つのボールをそれぞれ受けとめてください。――猫のチュー公，Pio，12月号，125―126pp.

MZ-80B/2500

▶ゼッター2
迷路の中にあるドットをすべて取るのだ。――黒武者健一，Pio，12月号，92―99pp.

MZ-2200

▶キャリングフルーツ
たかチャンを操作して，リンゴを4個すべて，小屋まで運べば1面クリアです。――児玉忠士，マイコン BASIC Magazine，12月号，99―100pp.

MZ-2000/2200

▶C・GRACE
バグ・プログラムの訂正とCGデータ。――編集部，テクノポリス，12月号，122―132pp.

▶CANDIES PANIC
キャンディが溶ける前にすべて食べてしまうゲームです。――登坂裕之，マイコン BASIC Magazine，12月号，96―98pp.

MZ-80B/2000/2200

▶不老長寿の林檎
リンゴをパクパク喰べたら，悪魔がやってくるゾ。――小畑健一，マイコン BASIC Magazine，12月号，94―95pp.

MZ-2500

▶NOBO
スクロールの滑らかなシューティング・ゲームです。――エンジンルームK.S，I/O，

12月号，247—260pp.
▶MZ-2500全回路図
全回路図公開。——編集部，I/O，12月号，329—349pp.
▶プリンタ便利化プログラム4
プリンタ関係のユーティリティプログラムを紹介する。——野瀬忍，マイコンライフ，12月号，104—109pp.
▶スーパーMZの内部解析
BASIC M-25についての内部解析——高橋雄一，マイコン，12月号，217—223pp.
▶MZ-2500使用記
MZ-2500の簡単な仕様紹介です。——編集部，Pio，12月号，170p.
▶MZ-2500ハードウェア解説
文字表示についてレポート。——編集部，ASCII，12月号，216—217pp.

共通

▶ヒーロー伝説
平和な国を取り戻すために，君が勇者となってバルードを倒すというロールプレイ・ゲームです。——馬場光男，マイコン，12月号，282—288pp.

X1/C/D/F/turbo

X1シリーズ

▶DISK BACK UP MAKER X1
プロテクトのかかっていないソフト，3インチ，5インチのメディア・コンバータとしても使える。——竹谷康範，I/O，12月号，291p.
▶PCGエディタ
ゲーム・プログラマのためのキャラクタ・エディタです。——ゲッピ，I/O，12月 298—303pp.
▶えっ!! X1/turboで「ファミコン」のプログラムが作られる？
ファミコンとX1との組み合わせで作る風変わりな開発ツール。——ホームコンピュータ研究会，マイコン，12月号，407—411pp.
▶サーチ・ラブ
チャックンポップのようにペンタは愛のためにムッチャンのハートを取り戻しに行くのだったあ～。——星合健二，POPCOM，12月号，179—183pp.
▶ドロボーくん
ドロボーくんの行く手を阻む悪魔達。お札を取って面クリア！——石木宏明，マイコン，12月号，270—274pp.
▶サウンドポプコム　タッチのテーマ
タッチのテーマ曲です。——越姫優香，POPCOM，12月号，51—52pp.
▶スペース　キャリア
TAGを操作して，誘導ロケットを運び，スペースシップを惑星まで届けてください。——Luck & Think，POPCOM，12月号，185—187pp.
▶円丈のジョ～ダンソフト
オトーサンをソウサして1丁目の駅から4丁目の自宅まで連れてくる，という単純なゲーム。——三遊亭円丈，POPCOM，12月号，107—112pp.
▶マイコンビジネスBASIC教室　ランダムアクセスファイルの活用
データベース型のファイル管理と，管理資料作成プログラムについて。——久安勝，マイコン，12月号，315—319pp.
▶ファイアレス　キャッスル
姫を取り戻すために王子はファイアレスに乗り込むのだった。——日野義政，マイコン BASIC Magazine，12月号，140—142pp.
▶水戸黄門
例によって，例のごとく助さん格さんは黄門様を守るのであった。——M.Y，マイコン BASIC Magazine，12月号，138—139pp.
▶ギルミス
ガルーダにさらわれた恋人を救うゲームです。——内海達哉，Pio，12月号，36—43pp.
▶もくじけん
少林寺の木人と戦うことになってしまった。——足立博之，Pio，12月号，127p.
▶フレックス
ギャラクシアンタイプのゲームです。——プロッタプリンタあ，Pio，12月号，126—127pp.
▶忍カンタン改造法　ギャラガ
面進め＆効果音遊びの隠しコマンド。——ケン・パックマン，テクノポリス，12月号，32p.
▶忍カンタン改造法　ニュートロン
フルーツひとつ取れば，面クリア。——えっくすわんZ，テクノポリス，12月号，32p.
▶C・GRACE
バグ・プログラム訂正，CGデータ。——編集部，テクノポリス，12月号，122—132pp.
▶すぐに遊べるマシン語入門
ゲームを作りながらマシン語に換えてゆく。ドライブゲーム編。——岡田一星，テクノポリス，12月号，110—120pp.
▶レーシング　コンストラクション
コースを作って，さあ走ろう！ゲームです。——酒井潔，テクノポリス，133—137pp.
▶SHARP X1用turbo LOGO
漢字版ロゴについてレポート。——編集部，ASCII，12月号，230p.

X1 turbo

▶スーパーパソコンX1 turbo II新登場
独創的なソフト2種が同梱されたハイコストパフォーマンスマシン，X1 turbo IIについて。——岡本一郎，マイコン，12月号，212—216pp，309p.
▶Big New Products X1 turbo model 31 & turbo LOGO
ニューモデル31には画期的なソフト2種が装備され，日本語処理機能がますます充実した。——編集部，I/O，12月号，243p.

ポケコン

PC-1245

▶PC-1245モニタ
8バイト，4バイト単位のモニタチェックが出来ます。——渡辺学，POPCOM，12月号，131—132pp.
▶ロイヤルポーカー
ポーカーゲームです。——高橋裕，Pio，12月号，150p.
▶墓参りゲーム
墓場のお化けにメゲずに宝を奪え！——辻，清水，Pio，12月号，148—149pp.

PC-1245/51

▶まくら投げ
先生の言いつけは素直に破って，夜は楽しくまくら投げ！——綾部鉄之介，マイコン BASIC Magazine，12月号，146p.

PC-1251

▶スクランブル
障害物をよけるスクロール・ゲームです。——IKSU，Pio，12月号，134—135pp.
▶蚊スペシャル
蚊を叩きましょう！——皿海みつる，Pio，12月号，133p.
▶ザ・モグラ
モグラ叩きゲームです。——伊藤正明，Pio，12月号，136—137pp.
▶ポケコン　ワード・ジェネレータ
ポケコンの1B出力を利用した4チャンネルのミニ・ワード・ジェネレータを製作する。——サタンゴースの息子マッドギャラン，I/O，12月号，224—227pp.
▶ヒデちゃんスイマー
川の中をスイスイ泳ごう。障害物には気をつけて。——栗生沢寿郎，POPCOM，12月号，134—135pp.
▶ポケコン・デジボル
OPアンプを使ってペン書きレコーダの代わりに，時間を横軸にとり，電圧の変化を記録するものです。——前原秀明，I/O，12月号，228—229pp.

PC-1245/50/51/55

▶スネーキー
大蛇が群れをなしてやってくる。ビーム砲でやっつけよう。——turboくん，Pio，12月号，146—147pp.

PC-1261

▶10×14ドット・キャラクタ出力ルーチン
そのような出力ルーチンです。——某中在GON，Pio，12月号，143p.

PC-1270

▶らんだむふぁいる　仕事に合わせてソフト選び　ポケコンPC-1270
仕事に合わせてROMカードの交換によってプログラムが使用可能なプログラム実行専用のポケコンPC-1270を発売。——編集部，POPCOM，12月号，102p.

PC-1350

▶ベースボール・ジャンケン
野球拳です。——3A3U，Pio，12月号，138—139pp.
▶ハンプンストンウェル物語
黒玉，セスナにぶつからないようにロケットを移動させてください。——ニョンタ，Pio，12月号，140—141pp.
▶惑星破壊ミサイルを迎撃せよ
3-Dワイヤーフレームによる迎撃ゲームです。——木本ごう，Pio，12月号，142—143pp.
▶ポータブル・ヤキュウケン
野球拳です。——Tk-SOFT KATAE俊行，Pio，12月号，144—145pp.
▶PC-1350情報
アドレスについての情報とPC-1350とPC-1401パスワード表示プログラム——清水徹，POPCOM，12月号，133p.
▶上・下スクロール
上・下スクロールプログラムです。——アパラチア・モドキ，POPCOM，12月号，134p.
▶ランディング・シミュレーション
フライト・シミュレーターです。RAMカード要。——チェスナット原，I/O，12月号，310—311pp.

PC-1500

▶キッチン　クルクル
キッチンにあるものが，日頃のうらみを込めて攻めてきた！——すっさん，マイコン BASIC Magazine，12月号，147p.

Info MZ

infoMZでは，小誌関連各社のご協力により，資料提供サービスを行っています。
毎月６社まで請求できます。FILES Oh！MZとあわせてご利用ください。

あ

アーバン電子㈱　〒730 広島市中区大手町1-1-23　☎0822(46)0993
㈱アートシステム　〒520 大津市木下町7-27　☎0775(24)0648
㈱アイテム　〒251 藤沢市南藤沢8-1-202　☎0466(27)1668
アスキー　〒107 港区南青山5-11-5 住友青山ビル　☎03(486)7111
㈱アテック　〒550 大阪市西区南堀江1-11-1　☎06(543)4051
㈱アバロン　〒150 渋谷区桜が丘町8-17 シャーレ渋谷Ａ棟102　☎03(476)0187
アビコン　〒590 堺市錦野町東1-1-6　☎0722(68)3399
池上通商㈱　〒101 千代田区外神田3-7-14 中部ビル　☎03(257)1685
㈱インテック　〒151渋谷区幡ヶ谷3-37-18　☎03(320)2110
ウェッジソフト　〒581 大阪府八尾市中田4 125 1　☎0729(91)6446
㈱ウス井パソコンセンター　〒930 富山市総曲輪3-6-3　☎0764(21) 4181
ウラカワ電器店　〒863 熊本県本渡市亀場町亀川1693-1　☎09692(3)2813
エイト電気㈱　〒110 台東区上野5-3-4　☎03(831)5632
㈱エース事務機　〒812 福岡市博多区店屋町2-11　☎092(291)5587
エデュカ㈱　〒160 新宿区内藤町1番地 三洋ビル　☎03(352)8611
㈱エニックス　〒160 新宿区西新宿7-1-6　☎03(366)4345
エプソン販売㈱　〒160 新宿区西新宿2-4-1 新宿NSビル私書箱6109　☎03(348)6801
㈱エムエーシーコンピュータ事業部　〒530. 大阪市北区曽根崎2丁目2-15　☎06-315-0541
応用電算技研㈱　〒810 福岡市中央区大名2-6-1 福岡国際ビル6F　☎092(715)2789
㈱オオカド　〒520-23 滋賀県野州部野州町小篠原2114　☎0775(87)1215
㈱オーク　〒615 京都市西京区川島有栖川町51 オークビル　☎075(391)0391
オークシステムハウス　〒772 徳島県鳴戸市撫養町斉田字大提18-6　☎08868(6)0105
オーディオイワサキ　〒073 北海道滝川市有明町2-1-44　☎0125(24)8772
オスコン電子㈱　〒150 渋谷区宇田川町41-1 NHK共同ビル　☎03(462)5013

か

加賀電子㈱　〒113 文京区湯島3-14-2　☎03(836)5161
カスタムデバイス社　〒323 栃木県小山市城東4-9-24　☎0285(25)3952
キャリーラボ　〒862 熊本市大江6丁目25-25 金子ビル1F　☎096(363)0211
ゲンコウシステム開発部　〒648-01 和歌山県伊都郡九度山町九度山1580　☎073654-2130
㈱光　栄　〒223 横浜市港北区日吉本町1876 光栄ビル1F　☎044(61)6861
㈱コスモスコンピュータ　〒164 中野区中野5-52-15 中野ブロードウェイ2F　☎03(385)5388
コスモス新潟　〒950 新潟市花園1-6-30 大幸ビル1F　☎0252(41)5721
㈲コマキシステム研究所　〒622 西宮市松生町14-18　☎0798(72)3514
㈱コンピュータアプリケーションズ生産技術研究所
　〒101 千代田区一ッ橋2-3-1 小学館ビル　☎03(263)7241
コンピュータシティ　〒640 和歌山市黒田255-1　☎0734(73)6553
コンピュータピング　〒013 横手市八幡石町27　☎0182(33)4767
㈱コンプティーク　〒102 千代田区隼町3-19 清水ビル4F　☎03(234)8041

さ

㈱ザインソフト　〒676 兵庫県高砂市米田町米田1162-1　☎0794(31)7453
堺教材センター　〒590 堺市櫛屋町4-2-16　☎0722(22)7148
㈱サムシングクット　〒160 新宿区大久保2-5-20 シティプラザ新宿3F　☎03(232)0801
サンワサプライ㈱　〒700 岡山市下中野801-4　☎0862(41)6336
三和電子㈱　〒173 板橋区中丸町58-5　☎03(955)9231
㈱シーティーエー　〒165 中野区野方1-53-5　☎03(388)3521
㈱シーレックスサンクラス　〒910 福井市二の宮5丁目14-3　☎0776(25)2111
㈱ジェーディーエス　〒156 世田谷区経堂2-1-26 小田急経堂ビル623　☎03(428)3359
㈱システムサポート京都　〒600 京都市大京区西洞院四条下ル 光悦ビル2F　☎075(343)4728
システムハウスコマンダー　〒461 名古屋市東区芳野1-18-9　☎052(932)1740
システムハウスMASH北川電子　〒471 豊田市若宮町5 47　☎0565(31)7644
㈱シャノアール　〒143 大田区山王3-29-1 フルク山王1F　☎03(778)0445
正 智 塾㈱　〒572 寝屋川市成田西町6-30　☎0720(34)6776
新紀元社　〒160 新宿区新宿4-1-9 新宿ユースビル　☎03(350)0848
シンキングラビット　〒665 兵庫県宝塚市中州1-2-24-503　☎0797(73)3113
新電子システム㈱　〒830 福岡県久留米市通東京3-4　☎0942(39)2404
数 研 塾　〒411 静岡県三島市一番町15-32 芹沢ビル4F　☎0559(75)1534
スカヤ無線商会　〒416 静岡県富士市長通104-3　☎0545(61)1417
ストラットフォート㈱　〒336 浦和市南浦和2-36-15　☎0488(85)5222
スヒタル産業㈱　〒101 千代田区外神田1-16-1　☎03(251)2918
㈲ソ ヒ ア　〒101 千代田区神田駿河台2-1 プラザお茶の水304　☎03(293)9407
ソフトウェアインターナショナル㈱　〒107 港区南青山2 9 28　SIビル　☎03(479)7151
㈱ソフト工学研究所　〒101 千代田区外神田3-11-2 ロックビル　☎03(251)1195
ソフト集団らくだ　〒720 広島県福山市引野町2-194　☎0849(43)6565
ソフトハウスBRAMS　〒177 練馬区東大泉4丁目22-16　☎03(923)8352
ソフトプロ㈱ソフトプロインターナショナル事業部
　〒530 大阪市北区西天満6-7-2 梅新東ビル5F　☎06(363)1221

た

㈱タイアモント社情報局DSL　〒100 千代田区霞が関1-4-2　☎03(504)6236
㈱タイエレクトロニクス　〒768 香川県観音寺市古川町切石211　☎08752(5)1308
タイナックス　〒336 浦和市北浦和4-5-8 吉田ビル☎0488(32)2356
㈱高木学園OAテック　〒444 岡崎市上六名3-13-2　☎0564(53)9400
チャンピオンソフト　〒530 大阪市北区西天満6 1 12　☎06(365)9900
ティーアンドイーソフト　〒465 名古屋市名東区豊が丘1810　☎052(773)7770
データウエストジャパン
　〒556 大阪市浪花区灘波中2-7-17 city松本201　☎06(632)0937
㈱データポップ　〒107 港区赤坂2-13-19 多聞堂ビル4F　☎03(584)1522
デービーソフト㈱　〒060 札幌市中央区北三条西2丁目 カミヤマビル　☎011(251)7462
㈱テクノソフト　〒857 佐世保市福石町4-14　☎0956(33)5555
㈱デジック　〒790 松山市本町6-6-7 ロータリー本町1F　☎0899(24)0914
テレシステムズ　〒556 大阪市浪速区恵美須1-2-5　☎06(631)0925
天昇電子㈱　〒101 千代田区神田須田町1-13-9　☎03(253)3947
㈱東海クリエイト　〒108 港区三田3-1-7 三田東宝ビル4F　☎03(456)4610
東京電子科学機材㈱　〒101 千代田区外神田2-2-12 神井ビル　☎03(257)1361

な

南 江 堂　〒113 文京区本郷3-42-6　☎03(811)7234
㈱日本コンピュータ設計　〒150 渋谷区松涛1-4-9　☎03(466)6101
日本統合ソフトウェア㈱　〒102 千代田区九段南2-3-14 靖国九段南ビル7F　☎03(263)2291
日本システム管理㈱　〒815 福岡市南区向野1丁目19-8　☎092(511)6932
日本ファルコム㈱　〒190 立川市曙町1-19-3　☎0425(27)4121
日本マイコン販売㈱　〒530 大阪市北区中崎西1-4-22 第8新興ビル　☎06(374)0849
日本ユニバック情報システム㈱OA事業部
　〒107 港区赤坂2-17-22 赤坂ツインタワー本館　☎03(587)8111
㈱ニデコ　〒101 千代田区神田松永町1　宮沢ビル3F　☎03(253)0761
ノグチコンピュータサービス㈱
　〒336 浦和市常盤9-20-3 北浦和第2大栄ビル　☎0488(31)0108
野島電機　〒670 姫路市今宿3-6-3　☎0792(92)0647

は

パーソナルメディア㈱　〒141 品川区西五反田8-1-11　☎03(490)8841
㈱パールビジネス　〒600 京都市下京区綾小路室町西入ル アネックス京都7F-B　☎075(371)8321
㈱パスカルII　〒432 浜松市佐鳴台4-9-10　☎0534(53)6186
㈱ぱ そ る　〒100 千代田区永田町2-4-3 永田町ビル1F　☎03(501)2181
㈱服部セイコー電子機器事業部
　〒104 中央区京橋2-4-12 京橋第一生命ビル　☎03(274)2121
ハネロン電子工業㈱　〒581 八尾市南木ノ本2-45-1　☎0729(94)7778
パル学習教室　〒530 大阪市北区天満1-7-17　☎06(352)0427
㈱パルコム　〒530 大阪市北区梅田1-1-3-900 駅前第3ビル9F　☎06(341)3422
ビービーエス　〒221 横浜市神奈川区西寺尾1-20-12 清水ビル2F　☎045(421)7421
ビクター音楽産業㈱　〒150 渋谷区渋谷1-7-5 青山セブンハイツ605　☎03(486)9470
福井コンピュータ販売㈱　〒910福井市二の宮3-42-30　☎0776(26)3430
㈱富士音響マイコンセンターRAM
　〒101 千代田区外神田1-15-16 秋葉原ラジオ会館　☎03(255)7846
プログラム企画サービス㈱　〒540 大阪市谷町1-44 大手前愛晃ビル2F　☎06(943)6344
ボーステック㈱　〒107 港区南青山5-12-3-204　☎03(407)4191
㈱北陸バイトショップ　〒930 富山市五福五区3216　☎0764(33)5176
㈱ホットビィ　〒164 中野区東中野4-4-1 丸新ビル5F　☎03(360)3623
㈱ポニーポニカ企画部　〒102 千代田区九段北4-1-3 日本ビルディング　☎03(265)6377
ポプコン　〒101 千代田区神田神保町3-3-7 昭和第2ビル4F　☎03(263)6940

ま

マーベルコンピュータ　〒673 明石市西明石南町1-10-13　☎078(923)5536
㈱マイクロキャビン　〒510 四日市市鵜の森1-2-15 メゾンヴァンベール2F　☎0593(51)6482
㈱マイクロソフトウェア アソシエイツ
　〒107 港区南青山7-8-1 小田急南青山ビル4F　☎03(486)1411
マイクロデータベース アソシエイツ　〒194-01 町田市鶴川2-1-8-205　☎0427(35)1600
マイクロネット　〒064 札幌市中央区南10条西15丁目 ムラカミビル3F　☎011(561)1370
マイクロプロジャパン㈱　〒107 港区南青山2-24-15 青山タワービル7F　☎03(423)4901
マイクロポート　〒657 神戸市灘区船寺通り5-3-8　☎078(801)5181
マイコンランド 上田　〒386 長野県上田市中央西1-15-28　☎0268(24)3515
マイコンセンター ウエノ　〒420 静岡県鷹匠3-14-16 コハラビル6F　☎0542(47)6211
マイコンシステム企画　〒546 大阪市東住吉区湯里1-1-1 稲田ビル403　☎06(704)9923
マイコンハウス シナプス1　〒730 広島市中区立町3-19 きよめビル1F　☎082(245)8833
マイコンショップ BASIC HOUSE　〒320 宇都宮市桜3-2-17 太平ビル　☎0286(33)1994
㈱マイコンハウスSPS　〒960 福島市太平寺町の内5-3　☎0245(45)5777
宮崎マイコンショップ　〒880 宮崎市宮脇町89-4　☎0985(27)4326

や

㈱ユニオンプランニング
　〒141 品川区東五反田5 28 9　第3花谷ビル3F　☎03(444)4161
ユニバース　〒700 岡山市下中野519-1　☎0862(44)1176
吉喜工業㈱　〒992 米沢市城南1-6-24　☎0238(21)1147

ら

㈱ライフボート　〒101 千代田区神田錦町3-6　☎03(293)4711
㈱ラウンドシステム研究所　〒560 豊中市上野西3-2-25　☎06(849)6982
リードレックス　〒150 渋谷区宇田川町36-6 ワールド宇田川ビル　☎03(464)1241
㈱リットーミュージック　〒160 新宿区四谷1丁目5　新四谷駅前ビル7F　☎03(359)0266
㈱リバーヒルソフト　〒810 福岡市中央区大名2-10-4 シャンボール大名B-202　☎092(771)3217
㈱ローランドディージー㈱　〒433 浜松市高丘町1006-14　☎0534(37)2333
BIG HEAD　〒380 長野市柳町76　☎0262(33)1911
CATS EYE CO.　〒537 大阪市東成区中道2-5-15-511　☎06(941)6379
㈱HAL研究所　〒101 千代田区神田須田町2-6-5 OSビル　☎03(252)5561
I・Oデータ機器　〒920 石川県金沢市高岡7-22　☎0762(21)4812
M I A　〒150 渋谷区渋谷2-9-1 青山田中ビル　☎03(486)4500
M.T.K電子㈱　〒660 尼崎市昭和通四丁目120　☎06(413)0188
㈱SBCソフトウェア パソコン流通課　〒160 新宿区本塩町21　木田建設ビル5F　☎03(353)9241
SQUARE　〒223 横浜市港北区日吉本町1776-3F　☎044(63)6201
㈱SSKオフィスコンピュータ　〒792 愛媛県新居浜市繁本町4-14　☎0897(32)3211
STUDIO M　〒982 宮城県仙台市金剛2-9-27　☎0222(45)3998

資料請求券 Oh！MZ '86.1	資料請求券 Oh！MZ '86.1	資料請求券 Oh！MZ '86.1	資料請求券 Oh！MZ '86.1	資料請求券 Oh！MZ '86.1	資料請求券 Oh！MZ '86.1

この資料請求券は，Oh！MZ編集室あてではなく，あなたが資料を送ってほしいメーカーあてに送るものです。

燃えるような紅葉の季節は終わり，霧降はもうすっかり冬です。今年の紅葉は例年にもまして色鮮やかで，それは素晴しいものでした。東照宮界隈，いろは坂，中禅寺などの名所ばかりがきれいなのではありません。何でもないところ，フッと通りかかったところの紅や黄と常緑樹の緑，山を眺めると自然の創ったモザイク模様。今年はこの美しさを存分に味わいました。早朝“ツナギ”を着てバイクで出かけたりしました。そうなんです！　ラッキーにもツナギが手に入ってしまったのです。友だちのお兄さんがクリーニング屋で縮められてしまったのを貰ったのです。それが私にピッタリ。フフ，あったかいし，たとえ転んでも守られているという安心感。それより，マシンと一体化して完璧にライダーになったんだ，という気がして。嬉しくて，近くでもわざわざツナギを着ています。でも，町中では目立ってしまって（そういう気だけ？）緊張してしまう。

ところで「風します」という漫画知ってます？『ビッグコミックスピリッツ』誌に連載中の４コマ漫画です。女の人が描いているバイクコミックで，“月子さん”というズッコケライダーが主人公です。思わずフッフと笑ってしまったり，フームと感じてしまったりします。ちょっと前に「創世紀」という題のものがありました。「初めにバイクがあった。犬，牛がまたがったが合わなかった。次に人間がまたがってピッタリだったので，バイクは人間のものになった」という話です。私はその話を読んで，バイクに乗れる人間で良かったなあ，と思ったものです。人間って，素晴しくできてるんですね。

それを改めてウーンと認識させてくれたのが，最近耳にするようになった“ファジィ理論”“ファジィシステム”です。ファジィ(fuzzy)というのは，fuzz（けば，綿毛）からきて「ぼやけた，あいまいな」という意味です。この理論で，人間は，はっきりしない情報をもとに判断して行動できるという素晴しい能力を持っている，ということに気づかされました。

いくら私のライディングテクニックが未熟だといっても，機械でバイクを運転するとしたら大変！　諸々の要素があるんだと思いますが，状況判断――行動ということを考えただけでも，なんと複雑。あいまいでしかも素早く的確なことをしていることか。まず，状況の情報を得るためのセンサーがすごいんでしょうね。視覚だけだって，あちこち前後左右，キョロキョロいろいろなものをとらえて選別して判断に導いている。道順，進路の決定，路面状況，周囲の交通，標識など，結構何気なくいろんな種類のことを見ています。おまけに右前方の車は初心者マークをつけた私の母ぐらいの年齢の人だから車間をあけて気をつけようとかも思ったりしています。ただし40km/h のときは停止距離22mだから，あと３m50cm車間をあけなければ，なんてしていません。こういう交通状況のときはこんなもんだ，という大体の状況把握と経験で行動しています。「だいたい」というのは細部の情報にはまるで正確さはないのですが，行動のもとになる全体的状況を素早くとらえるんですね。コンピュータはきっちりしていて０，１の２進数の組み合せで情報処理しているから「だいたい」とか「～ぐらい，という感じ」に弱かったんです。あるコンピュータ雑誌を見て知ったのですが，ファジィ理論を応用して地下鉄の自動運転システムや作業用ロボットが実用段階に入っているんですって。ロボットは，熟練した人間が経験から得ている判断行動を規則化して「もし～であれば…せよ」としているそうです。ところが「～」という箇所はいつも一定であるわけはない。「だいたい」というあいまいさを持つファジィ集合なんだそうで，ファジィ推論でこんなところだ，という具合にやらせたりするそうです。地下鉄自動運転システムなどは予見までするんですって。たとえば「快適に止まる」という目標があったら，現在速度の場合ブレーキ１回でこの距離で止まったらどうだ，３回に分けたらどうだ，と予見させて，いちばん指示目標に近い操作を最終的に行わせるそうです。なんと0.1秒ごとに最適な運転操作を決めているんですって。でも地下鉄は，他の交通がないからできるんでしょうね。なんだか，どんどん機械に人間の持つ「勘所」を教え込んでいるようですね。

バイオテクノロジー，バイオコンピュータ，人工知能などの研究が進み，先端技術ともてはやされたり夢を運んだり，また一方，脅威を呼び起こしたりしています。その開発結果に人間が征服され，支配されるようになったら怖いけど，「スターウォーズ」のR2―D2のように頑固一徹なロボットやC―3POのようにお調子者のロボットがいたら面白いな。人間は自分の手助けになるもの，代わりになるもの，それ以上のものを求め続けているんだと思います。でも結局「人間とは何か」という哲学的命題みたいなものに帰しているんじゃないでしょうか。

私は，最先端技術がもっとも古い学問と同じなのだ，というフレーズを頭に浮かべながら寒い風の中をバイクで，無意識のうちに予見行動しながら走りました。山は白くなっています。暮れてしまうと，ツナギを着ていても風が突き刺さってきます。部屋に戻り冷えた体を温めようと，白いコンピュータに向き合いジョイスティックを握り，ホットな気分になりました。

Q.A Oh!MZ 質問箱

Q **BASICで座標(39,24)に文字を表示したいのですが，どうしても改行されてしまってうまくいきません。「POKE $D3E7, n」以外に方法がないでしょうか。機種はMZ-1500です。**

愛知県　中野　智之

A 画面の右下すみに文字を表示すると改行してしまう。誰にも経験のあることでしょう。これに対処するには，直接VRAMに書き込む手があります。中野さんの質問中POKEというのがありますが，これがそうです。MZ-80K/C/1200/700/1500ならこれでよいのですが，MZ-80B/2000/2200やX1シリーズではIN, OUT, POKEを使い，複雑な処理をしてやらなければいけません。

さて，リスト1のプログラムはもっと簡単に実現する方法です。右下すみから1字左に文字を書き（これなら改行しない）同じ位置でインサートを行うことで，先ほど書いた文字をひとつ右へ送ってやれば完了です。80字モードのときも同じ要領で(78, 24)に書いてやればよいのです。10行〜30行中の自分の機種が書いている行と，40行から70行を打ち込んで試してみてください。

Q **Oh! MZ84年7月号にモニタの分離とその活用が出ていました。使用してみてたいへんよかったのですが，画面が40文字になっていてすこし見づらいので，これを80文字で使うようにするにはどうしたらいいでしょうか。　鳥取県　植田　操**

A 84年7月号に載ったのはX1についてですが，85年9月号には「S-OS番外地」にX1，MZ-80B/2000/2200のモニタの分離方法について書かれています。

まず，X1のモニタから80文字モードで使うには，単に「G098C」だけですみます。逆に40文字モードにするには「G0998」です。

MZ-80B/2000/2200で80文字モードにするには0C7CH番地，40文字モードにするには0CEEH番地をそれぞれコールすればよいのですが，モニタからのJ命令は「ジャンプ」であって「コール」ではないので，6バイトのプログラムを書き込む必要があります（S-OSのモニタからは「J0C7C」，「J0CEE」でOKです）。

80文字モードにするには，適当な（空いている）アドレスから

CD 7C 0C C3 B1 00

と6バイトのプログラムを打ち込み，その番地（たとえば12A0H番地）にジャンプすればよいのです。逆に40文字モードにするには6バイトのプログラムを

CD EE 0C C3 B1 00

とします。12A0H，12A6Hからそれぞれ書き込んで，モニタと一緒にセーブしておくと便利でしょう。（高野　庸一，泉　大介）

Q **X1のユーザーです。日頃，キーボードからダンプリストを打ち込むときに思うのですが，アルファベットのA〜Fの入力をテンキーの[/]，[*]，[-]，[+]，[=]，[.]を押すことでできれば速く入力できると思います。もしソフト的な方法があれば教えてください。　宮城県　横山　義行**

A X1で16進キーを実現するには，IOCSを書き換えて一度ほかの機械語ルーチンへ飛ぶ必要があります。そのためのプログラムがリスト2−A，ソースリストがリスト2−Bです。ただし，8CB01，8FB01用（NEW BASICも可）です。turbo BASICでは動きません。

使い方は，モニタから「GE000」もしくはBASICで「CALL &HE00」です。テンキーの[/]，[*]，[-]，[+]，[=]，[.]がそれぞれA〜Fになり，[.]がスペースになります。もし[.]をスペースにしたくない場合はE049H番地を2CHにしてください。また，元に戻すには「GE002」です。

さて，このプログラムは簡単にリロケート（移動）することができます。たとえば，C000Hへ移動する場合は，リスト2−AをC000H番地へこのまま打ち込んだあとで，E005H番地からの2バイトを00H，C0Hにしてください。16進キーモードにするには「GC000」で，元に戻すには「GC002H」です。

（高野　庸一）

Q **最近S-OSを打ち込んでいて実感したことです。マシン語を打ち込むときA〜Fを打ち込むのに骨が折れます。そこで，テンキーを16進キーにしたいのですが，マシン語は始めたばかりでよくわかりません。僕のマシンMZ-2200にはテンキーの部分に全部で15個しかキーがありません**

リスト1　画面右下すみに文字を表示

```
10 INS=24  : ' 80K/C/1200/700/1500 ノ S-BASIC
20 INS=8   : ' 80B/2000/2200       ノ S-BASIC
30 INS=18  : ' HuBASIC
40 CURSOR 38,24 : PRINT "A";
50 CURSOR 38,24 : PRINT CHR$(INS);
60 CURSOR 0,0
70 END
```

リスト2-A　16進数キーダンプリスト(X1用)

```
E000 18 02 18 0C 21 00 E0 11 :50
E008 1E 00 19 EB 3E C3 18 05 :40
E010 3E 7A 11 E6 02 21 B4 03 :89
E018 77 23 73 23 72 C9 7A E6 :CB
E020 02 CA B7 03 7B 1E 41 FE :5E
E028 2F 28 20 1C FE 2A 28 1B :FE
E030 1C FE 2D 28 16 1C FE 2B :CA
E038 28 11 1C FE 3D 28 0C 1C :E0
E040 FE 2E 28 07 FE 2C 20 02 :A7
E048 3E 20 5F C3 B9 03       :3C
---------------------------------
SUM: 9C EE 5C 0F 56 68 B9 61 :CD
```

が，たとえば，SHIFTキーと[0]を押したら「A」になるようにすればよいと思います。

兵庫県　田中　透

A テンキーとシフトキーを併用して16進キーにするというのは，おもしろいアイデアですね。リスト３－Ａのようなプログラムを作ってみました。8000$_H$にジャンプすればテンキーが16進キーになります。[SHIFT]＋[0]でA，[SHIFT]＋[5]でFが出ます。Aは10，Fは15ですから覚えやすいでしょう。[SHIFT]＋[0]が押しづらいようなら[SHIFT]＋[7]でも結構です。

この機能をクリアするには8002$_H$へジャンプしてください。このプログラムは，モニタ1Z001Mの内部ルーチンを変更後モニタへジャンプするようにできています。モニタは1Z001Mでなければ動きません。

ほかのアドレスにこのプログラムを動かしたいときはリスト3-A中□で囲ってあるところを次のように書き換えてください。たとえばFE80$_H$へ動かす場合は次のようにします。

12　80 → 92 FE（$92_H = 12_H + 80_H$）

20　80 → A0 FE（$A0_H = 20_H + 80_H$）

このプログラムはほかのプログラムから壊されることのないアドレスに置くようにしてください。さもないと暴走しますよ。

（泉　大介）

お詫びと訂正

Oh！MZ 85年10月号 167ページ質問箱で，加藤孝治氏の名前が「加藤考治」となっていました。訂正してお詫び申し上げます。

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力をあげてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に，機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒102 東京都千代田区四番町2-1
(株)日本ソフトバンク
出版部「Oh！MZ質問箱」係

リスト２-Ｂ　16進キーソースリスト（X1用）

```
 1 E000                    ORG     0E000H
 2                 ;
 3 E000 1802       HEXKEY: JR      PATCH0
 4 E002 180C       NMRKEY: JR      PATCH1
 5 E004 2100E0     PATCH0: LD      HL,0E000H       ;!!!!!!
 6 E007 111E00             LD      DF,CONV-0E000H
 7 E00A 19                 ADD     HL,DE
 8 E00B EB                 EX      DE,HL
 9 E00C 3EC3               LD      A,0C3H
10 E00E 1805               JR      MOVE
11                 ;
12 E010 3E7A       PATCH1: LD      A,7AH           ;ORGIN. CODE
13 E012 11E602             LD      DE,02E6H        ;ORGIN. CODE
14                 ;
15 E015 21B403     MOVE:   LD      HL,03B4H        ;PATCH
16 E018 77                 LD      (HL),A
17 E019 23                 INC     HL
18 E01A 73                 LD      (HL),E
19 E01B 23                 INC     HL
20 E01C 72                 LD      (HL),D
21 E01D C9                 RET
22                 ;
23 E01E 7A         CONV:   LD      A,D
24 E01F E602               AND     2
25 E021 CAB703             JP      Z,03B7H         ;RETURN
26                 ;
27 E024 7B                 LD      A,E     ;E=ASCII CODE
28 E025 1E41               LD      E,'A'
29 E027 FE2F               CP      '/'     ;'/'->'A'
30 E029 2820               JR      Z,MATCH
31                 ;
32 E02B 1C                 INC     E       ;E='B'
33 E02C FE2A               CP      '*'     ;'*'->'B'
34 E02E 281B               JR      Z,MATCH
35                 ;
36 E030 1C                 INC     E       ;E='C'
37 E031 FE2D               CP      '-'     ;'-'->'C'
38 E033 2816               JR      Z,MATCH
39                 ;
40 E035 1C                 INC     E       ;E='D'
41 E036 FE2B               CP      '+'     ;'+'->'D'
42 E038 2811               JR      Z,MATCH
43                 ;
44 E03A 1C                 INC     E       ;E='E'
45 E03B FE3D               CP      '='     ;'='->'E'
46 E03D 280C               JR      Z,MATCH
47                 ;
48 E03F 1C                 INC     E       ;E='F'
49 E040 FE2E               CP      '.'     ;'.'->'F'
50 E042 2807               JR      Z,MATCH
51                 ;
52 E044 FE2C               CP      ','
53 E046 2002               JR      NZ,NOCNGE
54 E048 3E20               LD      A,' '   ;','->' '
55 E04A 5F         NOCNGE: LD      E,A
56 E04B C3B903     MATCH:  JP      03B9H
57                 ;
58 E04E                    END
```

リスト３-Ａ　16進キーダンプリスト（MZ-2000/2200用）

```
8000 18 02 18 34 21 B4 09 36 :7A
8008 C3 21 12 80 22 B5 09 C3 :19
8010 B1 00 79 FE 02 D2 B8 09 :BD
8018 78 FE 04 C2 CC 09 7B FE :8A
8020 10 DA CC 09 3E 17 BB DA :A9
8028 CC 09 21 20 80 19 7E C9 :F6
8030 41 42 43 44 45 46 36 41 :0C
8038 21 FE 02 22 B4 09 21 38 :59
8040 14 22 B6 09 C3 B1 00    :69
--------------------------------
SUM: 56 66 8F 0C 8B 74 D5 1C :47
```

リスト３-Ｂ　16進キーソースリスト（MZ-2000/2200用）

```
0000                 1 ;
0000                 2 ;   EXTENDED KEY ROUTINE
0000                 3 ;
0000                 4
8000                 5          ORG     8000H
8000                 6
8000 18 02           7          JR      PATCH
8002 18 34           8          JR      CLCT
8004                 9          ;
8004 21 B4 09       10 PATCH:   LD      HL,09B4H
8007 36 C3          11          LD      (HL),0C3H
8009 21 12 80       12          LD      HL,START
800C 22 B5 09       13          LD      (09B5H),HL
800F C3 B1 00       14          JP      00B1H
8012                15          ;
8012 79             16 START:   LD      A,C
8013 FE 02          17          CP      2
8015 D2 B8 09       18          JP      NC,09B8H
8018 78             19          LD      A,B
8019 FE 04          20          CP      4
801B C2 CC 09       21          JP      NZ,09CCH
801E 7B             22          LD      A,E
801F FE 10          23          CP      15+1
8021 DA CC 09       24          JP      C,09CCH
8024 3E 17          25          LD      A,17H
8026 BB             26          CP      E
8027 DA CC 09       27          JP      C,09CCH
802A 21 20 80       28          LD      HL,TABLE-16
802D 19             29          ADD     HL,DE
802E 7E             30          LD      A,(HL)
802F C9             31          RET
8030                32
8030 41 42 43 44    33 TABLE:   DEFB    'A':'B':'C':'D'
8034 45 46 36 41    34          DEFB    'E':'F':'6':'A'
8038                35
8038                36
8038                37 ;
8038                38 ;   ﾓﾄﾆ ﾓﾄﾞｽ
8038                39 ;
8038                40
8038 21 FE 02       41 CLCT:    LD      HL,02FEH
803B 22 B4 09       42          LD      (09B4H),HL
803E 21 38 14       43          LD      HL,1438H
8041 22 B6 09       44          LD      (09B6H),HL
8044 C3 B1 00       45          JP      00B1H
```

STUDIO MZ

◆11月号のマシン語特集を読んでぼくの体の中では炎がメラメラと燃えあがった。しかし10分もたたないうちにぼくは燃えつきてしまった。それは，ぼくが世にも恐ろしい○○コンだったのだ(ロリコンじゃないよー)。　松田　努 (14) 兵庫県

簡単に燃えつきないで火種だけは残しておいてくださいね。マシンを手に入れたときのためにも。

◆マシン語特集に「逆上と挫折の歴史」を書いていた工藤誠さん。あなたはまだ運のいい方ですよ。ぼくの場合なんかは，5月号のEDASを打ち込んでから，さあ使ってみようと思ってからふと気がついたんです。使い方がわからないことに。それからの約3週間，EDASというアドベンチャーに没頭し，やっとのことでAS，I，Cなどの命令が少しずつわかってきたのです。そしてなんとか自作のチェックサムが完成したときOh!MZのEDASMを見たのです。さすがに文字のMが多い分だけ使いやすく，独特のマクロも装備している。思わずぼくはEDASを捨ててしまいました。

高木　淳 (15) 東京都

◆マシン語特集はたいへん参考になりました。とくに「わが青春のマシンコード」を読んで，自分の未熟さに気付きました。実はぼくも最近マシン語を始めたばかりなのですが，その第1作として5Kバイトのゲームを製作中です。当然のことのように暴走の連続ですが，しかしぼくは絶対にあきらめない。　横田　紀明 (18) 山口県

◆11月号のマシン語特集は最高でした。いまひとつ理解できなかった部分がかなり解決しました。いまは，BASICで作ったプログラムをマシン語に作り替えています。今度，また投稿しますからお楽しみに。　宇治原　徹 (15) 三重県

楽しみに待ってます。

◆X1turboを手にしてから3カ月，ようやくBASICのBAの字ぐらいが見えてきました。それにしてもPCシリーズの本はごまんとあるのに，turboに関する本の少ないこと。だからOh!MZだけが頼りです。いまだに理解できないところもかなりありますが，自作プログラムを作って送れる日を夢見て，毎日キーボードに向かっています。これからもOh!MZを頼りにしていますので，よろしくお願いします。

三田　和英 (36) 千葉県

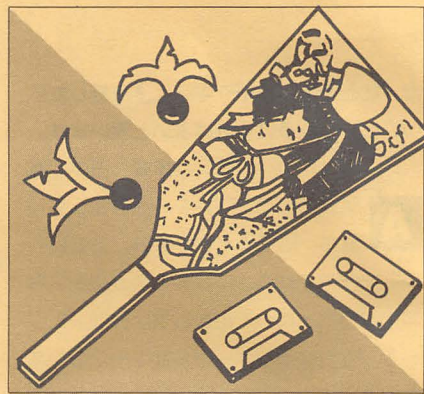
今月の特集で，IかCの文字まで見えてくるといいですね。

◆11月号のmicroOdysseyには感動した。特に「○○は○○らしく」の1文は，日ごろぼくが考えていることと同じである。もし「○○らしくない」ではどうだと言うのだろうか。けど，学生らしくないと言ってはずれるのもつまらない。まあ，気楽に生きようと思う。　川野　博司 (15) 大阪府

◆11月号のAgain Watchを読むまでは，モーターボートがほしいと思っていたが，買うのは当分あとになりそうだ。　足立　嘉孝 (17) 兵庫県

◆日経の「経済前線ランキング60年度版」を見るとおもしろいですよ。まず，59年のパソコンのシェアは，シャープは相変わらず3位ですが，58年に17%→10%と落ちていたシェアが10%に踏みとどまり，今年は上昇してくれるのではないかと期待させてくれます。ちなみに2位の富士通は5%増の17%，1位のNECは±0%で47%維持のままでした。パソコンはこれからという感じですが，日本語ワープロのほうは，シャープが6%増の14%で3位に上昇しています。ちなみに前年1位の富士通は11%もシェアを減らして24%で2位に落ち，代わっての1位は13%増のキヤノンで27%でした。さて，最後に最も注目すべき点はカラーテレビのシェアで，市場全体で−1.3%の成長率にもかかわらず，シャープのシェアは1%増の13.5%で3位に浮上しています。そしてこの原動力となったのが，われらがパソコンテレビX1なのでありました。結局これが言いたいだけだったりして……。ついでに家庭用VTRでも4位に浮上していました。シャープってエライ会社なんですね。

土居　秀二 (22) 京都府

このコーナーにもAgain Watchなみのお便りが届くようになって，たいへんうれしく思っています。

FROM READERS TO THE EDITOR

STUDIO MZは，皆さんからお寄せいただいている愛読者カードのメッセージをメインに構成されています。ここいちばんの本音を聞かせてください。あわせてイラストも募集します。ユニークなオリジナル作品をお待ちしています。

◆寒風吹きすさぶ中，愛車DT125とともに街をかけ抜ける。赤信号で止まりふと前を見ると，そこには赤のミラノX-1turbo。いつかは買ってやるぞX1turbo!　と思いつつ赤いミラノをブッチギった私であった。　山口　幸一 (19) 宮崎県

◆シャープ本社OAショールームのおねえさんはとても親切です。夕方だとほとんど誰もいないので，パソコンショップよりもずーっと落ち着いて，しかもていねいに教えてもらえます。この前MZ-1500でフラッピーをやらさせてもらっちゃった。

柴田　昌彦 (16) 大阪府

市ヶ谷のショールームも楽しいですよ。

◆ナッナント，私のクラスにはMZ，X1ユーザーを含めて9人ものパソコン少年がいる。最初Oh!MZを買っていたのは私と友人の2人だけだったのが，今では9人全員が読んでいる。“Oh!MZで広げよう友達のワッ!!”　阿部　崇 (15) 青森県

大きな輪にしていきたいですね，これからも。

◆10月10日：MZ-2500を買った。ひじょうにうれしい。10月12日：交通事故にあって入院した。ひじょうにイタイ。10月23日：病院のベッドでこのハガキを書く。ひじょうにくやしい。早く退院して，MZにほおずりしてナデナデしてやりたい。とにかくクヤシイ！　小岩　敬一 (24) 岩手県

◆ぼくらの掲示板のコーナーに，「差し上げます」とか「よろしければもらってください」という項目はないのでしょうか。私は人にプレゼントをして喜んでもらうのが大好きな性格なのです。もしそういうコーナーができればこの私にお任せください。　小島　敬子 (29) 愛知県

お嫁さんにもらってくださいっていうのもいいんでしょうか？

◆11月号STUDIO MZの大西さんはエライ！　ぼくもあの「近いうちにMZ-80Kの特集を必ず……」にダマされて，今まで待っていたひとりです。今こそMZ-80Kのパワーアップ講座を。

宮内　博文 (17) 広島県

◆中学校へ教育実習に行ってきました。そこで，「先生はMZ-1500を持っているのだ」と自慢すると，「ぼくは○○を持ってるよ」，「ぼくも△△を……」と，すごい機種の名前が返ってきました。最近の中学生はすごいんですね。

竹内　寛泰 (21) 広島県

マシンは“もの”ではありません。使ってこそのパーソナルコンピュータです。

◆タバコを1カ月吸わずにためたお金で，ゲームソフトを2本買って帰ったら，X1本体が動かなくなっていました。修理代はないし，タバコはやめたし，今度は何をやめればよいのでしょうか。

中村　一浩 (34) 福岡県

それはですね，タバコをやめたお金でまず修理に出して，しばらくはゲームをやめてしまうのがいいんじゃないんでしょうか。

◆私の会社に出入りのある某社の特約店のMさんに「MZ-2500を売ってください」と言ったところ，一瞬ズッコケたものの，気持ちよく承諾してくれました。ところが慣れない仕事をしたものだから，値引き率を間違えてしまったらしくて，会社の社長さんからしかられてしまったそうです。無理な注文をしてゴメンナサイ。

立川　賀規 (32) 埼玉県

◆「TOKYOナンパストリート」に登場する女の子には鼻がないのを知っていますか。地元から横浜までのタクシー料金は？　ディスコ，レストランと連れて行って，ホテルに誘うと補導員だったということありましたか？　レストランのトイレでオカマと……したことありますか？　うまく口説いたつもりがプロの女性で，怖いおにいさんにどつかれたことは？　ぼくはすべての経験者です。

石塚　孝幸 (18) 奈良県

私は彼女の父親に追い出されたことがあります。

◆ナンダ！　あの「TOKYOナンパストリート」というのは。やたら難しいじゃないか。よっぽど本物のほうが簡単だぜ！　(ウン，なかなか含みのある発言だ)　神崎　寛 (17) 広島県

その成果を報告していただきたいものですね。
◆「オオッ！ コンピュータが考えとる考えとる」と騒ぎながら，みんなで楽しめるソフトが「棋太平」じゃ。あの考えている(?)間がまた楽しい。しかし，そばで見ていた奥さんが一言。「最初からそのようにプログラムしとるんだよ」だって。女の人って現実的で，夢もチボウもないんですねェ。

華谷　均（26）愛媛県

◆「軽井沢誘拐案内」を解いた。これはひとりで自力で解いた最初のアドベンチャーゲームだ。このゲームはとても素晴しい出来で，なんとも言い表せない感動を覚えてしまった。もう最高。

田村　憲明（14）北海道

◆最近，ポータブルワープロがたくさん出まわっている。メーカー，機種ともに多様である。しかし，コンピュータと連動できるようにするともっと売れると思う。たとえば，プリンタや変換機能を利用して，コンピュータで処理できるようになればいいんじゃないでしょうか。

山田　慎（15）愛知県

そのとおり。メーカーさんも勉強しなくてはユーザーに馬鹿にされますね。

◆突然，父が「WD-600なるワープロを買うぞ」とのたまいました。「あんな高価な物をどうして？」と聞くと，「20万円近い値段で買えるからだ」と言いました。それでもかなり高いと思ったのですが，実際に店頭で使ってみるとなかなかのもので，今から早くこないかと待ちどおしい今日この頃です。

中川　英也（17）三重県

◆PITMANの移植版は勢ぞろいといったところですね。うちには兄のMSXしかないので，それに移植してみましたが，なかなか楽しめます。また，新しい面を発表してくれるのでしょうか。

伊藤　英彦（22）青森県

◆ぼくの家では，お父さんもOh!MZのファンなので，いつもぼくと取り合いになります。今度から2冊買おうと思います。

島津　泰信（14）神奈川県

◆教育ソフトの中に，地図帳などのソフトがあってもよいのではないかと思います。たとえばメニュー画面が日本地図や世界地図で，番号で地域を指定して，地名，国名，首都などを画面上に呼び出すことができ，人口や国王，大統領名は書き換えが可能なんていうのはどうでしょうか。

田原　健児（30）宮崎県

X1の「ユーザー辞書」を活用すれば簡単にできそうです。

◆阪神タイガースが日本一になり，まるで正月が21回来たようです。阪神タイガース日本一バンザーイ！　　森　晴信（15）京都府

◆lex・i・con[léksəkán, -sikən|-sikən]n. 1 ((特にギリシア語・ヘブライ語・アラビア語などの)) 辞書(dictionary)。2 ((特定の言語・作家・作品・分野などの)) 語彙(ごい)；語彙集；〔言〕語彙目録（研究社，新英和中辞典・第4版による）

宗片　陽一（16）山形県

試験にでも出るといいですね，この単語。

◆シャープから今度発売される「カラーイメージボード」はいいですね。いままで100万円以上したものが4万円とは安いですねー。

小野　秀貴（16）秋田県

◆大阪のエレクトロニクスショウ'85のシャープのブースで見た，MZ-6500を使った翻訳システムはすごかったですよ。ぜひOh!MZでも紹介してください。それと，X1用のカラーイメージボードも最高でした。　　麻田　幸男（16）京都府

◆私は毎月，4冊ものクルマ関係の雑誌を買っているあちらの筋の者ですが，それらの本を見ていて提案があります。つまりパソコンの新機種をスクープしてしまうのです。たとえば，このような記事です。――証言者のK氏は生々しく新機種の詳細を語ってくれた。「あのキーボードは5500や6500とも違う，確かにそれらよりリファインされている。まさしく7500に違いない」と――いかがでしょうか？　そうなれば，買ったすぐ後に新機種が出てしまうということがなくなるのでは。

畑　克海（19）神奈川県

車のようにアウトバーンやテストコースを走っているパソコンをスクープしたい。

◆11月号のSTUDIO MZにOut door lifeに関するお便りもたまにはいいと書いてあったので，このぼくが……。ぼくは高校を卒業したら絶対にHONDAバラードスポーツCR-Xsiを買います。ナント値段は150万円。ズズーン！　CR-Xsiの特長は，全長3675mm，全幅1625mm，全高1290mm，車両重量860kg，エンジン形式ZC型1.6ℓ DOHC，最高出力135ps/6500rpm，最高速度200km以上。ぼくはこのCR-Xsiにハードショック，強化パッドとZC1.7ℓ＋turbo＋インタークーラー（270ps）のエンジン積んで，さらに無限CR-X RROエアロできめて，サーキットからリーフィールドをかっとぶのだ。

永尾　博幸（17）兵庫県

カラーはシルバーメタリックがお薦め品だと思いますよ。

◆11月号から買い始めました。そこでZEDA，ZAID，MACINTO-Sなどをバックナンバーを注文して使いたいのですが，当然のことながら6月号のS-OS MACEがないと動きません。しかし6月号のバックナンバーが品切れとのこと，いったいどうすればよいのでしょうか。まだこれからもMACEを使ったプログラムが載ると思うんですが。

岸　稔（18）兵庫県

お待たせしました。来月発売の2月号にバージョンアップ版を掲載するので，ぜひ活用してください。

◆VP-80Kを買ったのですが，turbo用の設定がどうもうまくできません。どうしても漢字があると文字がずれてしまいます。誰かVP-80K用のプリンタ設定データを教えてください。

山口　順一（26）鹿児島県

VP-130KモードでOKのはずです。

◆フッフッフ，11月号のものすごーいミス（バグとも言う）見っけ。P.148「PITGAL」のプログラムの340行を見てください。MOVE MANと書いてあるのさっ。どーだ，まいっただろう。

浅辺　公彦（15）広島県

これはですね，あのキャラクタはスカートをはいた男の子だということでは……。

◆朝起きるとぼくは女の子になっていた。あたりを見まわすと金塊があったので取ろうとすると，石がじゃましてうまくいかない。それでもなんとかして全部取ると，新しくまわりの景色がかわった。永久に終わることがない部屋のなかで……と思っていたら夢だった。高橋　昇一（16）奈良県

夢でよかったのでしょうか。

◆PITGALを打ち込んで遊んでいます。これはフラッピーに匹敵する面白さですね。作者の方は高校生なのかな？　もしそうだとすれば，学生さんをやりながらこんなにも楽しくて難しいゲームを作るなんて素晴しいことですね。それにキャラクタもとってもかわいいですよ。

森田　光子（61）北海道

とても理解のあるお母さんの登場です。

◆11月号のZ80ニーモニック表を見て，思わず「バンザーイ！」と叫んでしまいました。だって2日ぐらい前にマシン語の本を読みあさって，「表はないのか，表は」とぼやいていたんですから。Oh!MZは，読者の声ならぬ願いを反映してくれるんですね。　　金田　真生（16）東京都

◆先日，turboの「システム辞書」を使って「夏目」と入力，その後「夏目漱石」と出るのを待った――が出てこない。やはり第2水準は必需品である。このままでは1,000円札に申し訳がたたないではないか。ところで三菱自工の製品の型式番号はどうなっているか知ってます？　ナントMZなのです。

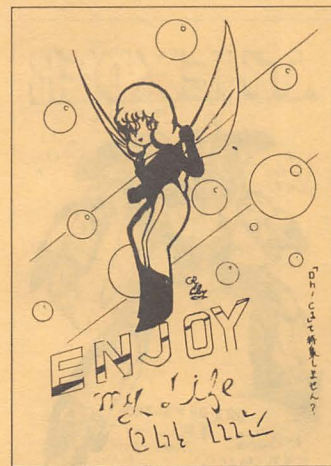
臼渕　啓省（19）東京都

◆私の父はとてもサンダーフォースが好きなのである。今のところ最高記録は，テープ版を2周ロードさせて57360点とすさまじい。ひょっとすると，うちの父は日本一サンダーフォースがうまいのではないだろうか。　松尾　匡章（16）兵庫県

それはすごい。

◆「火の鳥」のテレビCMはよく見かけるようになったが，しかしturboを忘れてもらっては困る。ここは一発『美味しんぼ』風に，「まあ！　Z80のくさみが全然なくて，漢字のいい香りがほんのりと……」，「高速グラフィックのうま味を引き出すのは，ごてごてしたインタプリタをのっけるだけが能じゃない。これはシャープの技術があればこそなんだが……」てなぐあいに，あのキャラクタを使ってみてはいかがでしょうか。

新居　弘司（22）大阪府

◆「ゲームばかりやっているけど，このままではいけない」と，いつも考えてはいるけれど，ついつい今日もゲームをやってしまって悩んでいる方々へ。もしまる一日でもフリーの時間がとれれば，パソコンの横に持っているゲームソフトを全部積み上げて，次の日の朝までブッ通しでゲームをやり続けてみてください。きっとコンピュータに対して「すまない，こんなにこき使って，ほんとうに悪かった」という気持ちでいっぱいになります。すると，それ以降はBASICをロードする割合がかなり増えるでしょう。BASICでプログラムを考えたり作ったりするのも楽しいですよ。8ビットだって，その気になればいくらでも使いこなせるの

河　明義　大阪府

です。「8ビットはゲーム専用」というのは，誰かが言った迷信なのです。
田村　佳則（19）埼玉県

パソコンの革命的な発展はいつも8ビットマシンからですよ。

◆今年の5月号の「買います」のコーナーを読んで，せっかく連絡してあげたのに，いっこうに返事をくれない人がいます。その後，何回か連絡しても何も言って来ません。もう以前のことなので，これ以上とやかく言いませんが，とにかく気分がすっきりしません。皆さんもこんなことのないように，ハガキを出すのなら必ず責任を持って対処するようにしましょう。加藤　雅勇（16）兵庫県

当然のことですね。これからハガキを出される皆さんはこんなことありませんよね。

◆パソコンも3台目となり，今も昔も変わらず子供とゲームばかりの毎日です。当初の目的ではCAIプログラム用にと思って買ったのですが，それもままならず，なんとかOh!MZで参考になる記事をお願いします。　大原　一男（32）広島県

今月の特集はいかがでしたか。

◆少し古い話ですが，今年の6月号の隠れプレゼントのNEW BASICはいったいどこにあったんでしょうか。いまだにぼくは見つけられません。誰か教えてください。　坂　直樹（13）三重県

◆少年は思った。「まいったぜ。奴らの進歩はすざまじい。やっとのことで追いつくと，奴らはまた新しい技で攻めてくる。今の俺なんかじゃとても追いつくことはできない。しかし今にみていろ俺だって……」と部屋に閉じこもり，秘かに復讐を誓う17歳の冬であった。弘　俊亮（17）山口県

◆夜勤のときでも，Oh!MZの発売日には昼にしっかり目を覚まして本屋さんに出勤します。毎月18日が楽しみです。　長田　光則（21）静岡県

◆私の身のまわりには，プログラムについて聞くことのできる人がいなくて困るのですが，Oh!MZの編集室ではバグに関する問い合わせにも快く応じていただいて，たいへんありがたく思っています。これからもよろしくお願いします。
寺川　雅元（52）高知県

◆11月号の「買います」のコーナーに載せていただいたのですが，多くの方々からお手紙をいただき驚いています。残念ながらすべての方々から購入するわけにもいかず，ほとんどの方におことわりの返事を書くことになってしまいました。この場をかりておわびとお礼を申し上げます。でもほんとうにMZファンの方々は善人が多いですね。遠くは九州からもお手紙をいただき，また多くの方が私の希望価格以下の価格をハガキに書いてこられ，私の方がなにかすまないような気分になってしまいました。MZはマイナーになったと考えていた私が間違っていたようです。今はMZよ永遠にといった気分ですね。　山本　直幸（23）茨城県

勝間田　裕　山口県

◆STUDIO MZの仲間募集のコーナーに，年齢も書くようにしてください。そうすればレベル的にも同じ人が集まるだろうし，私ももう年なので小中学生のサークルにはチョット……。
村田　嗣浩（25）北海道

これからは仲間募集のハガキを送ってくれる場合には平均年齢を書いてくれるとありがたいですね。

◆皆さん知ってますか？　ナ，ナント，サロンパスのTVコマーシャルで，泉ピン子のうしろにはあの名機MZ-2000が置いてあります。
小沼　勝則（15）茨城県

◆約2年半前に買った私のMZ-711は，今月とうとう721になりました。おそらく日本で最後にできた721だと思います。とにかく原始的だろうが，なんだろうが，カセットを制御するのに感動している最中です。あとはプロッタプリンタだけど，たぶん売っていないだろうな。
菅谷　薫（19）茨城県

◆最近，「受験のためにパソコンを人に貸す」などという軟弱者の多いのに私は失望している。私なんか，テニスの部活動には毎回必ず出席し，家に帰るとパソコンと寝るまでにらめっこ。それでも京都大学工学部電気工学科にみごとに現役合格してしまった……という夢を先日見たのだ。
酒井　英典（18）愛知県

この夢を正夢にするためにも，これからがんばってくださいね。

◆私は富田靖子ちゃんのファンです。もちろんOh!MZのファンでもあります。どちらもまだメジャーではないので，まだまだこれから発展性がありそうです。だから好きなのですOh!MZが。そこで相談ですが，表紙はカラーイメージボードで作ったグラフィックを使ったらいかがでしょうか。
猪口　正樹（17）佐賀県

◆10月21日(月)に共通一次試験の願書を学校に提出しました。それから英単語暗記ツールを作ったのですが，データ（単語数）が多いので入力がたいへんです。そこで試験に出る単語ディスク版なんてあればいいと思います。ところで，私は共通一次試験廃止には反対なのです。それはなぜかと言うと，私は理系でありながら国語，日本史のほうが数学や物理よりも得意なのです。したがって科目減にも反対なのです。
安孫子　朋之（17）茨城県

◆Oh!MZを完全読破すると体重が1kg減ってしまう。それほど中身が濃いのさOh!MZは。
青柳　伸之（17）千葉県

やっぱり。

◆先日，コム・ロードに行ったら「とうとう付けてしまいました防犯カメラ」と書いてあった。
竹村　敏広（17）愛知県

でも何故なんでしょうね。

◆とうとう裏表紙のMZ-5500/6500の広告がレギュラー落ちですね。'84年5月から18カ月は大記録でしょう。この記録を抜くのはどの機種なんでしょうね。　都成　孝之（16）大阪府

◆タイガースがリーグ優勝したので，私はもちろんのこと，家内も有頂天でした。このチャンスを逃すまいと「Super MZがほしい」と言ったら，あっさりと「買えば」と答えてくれた。しかし敵もさるもの「電子レンジを買い替えたい」と言われてしまった。そこでニコニコしながら「シャープのにしなさい」と言ってやったんです。家内はさっそく次の日曜日に買いに出かけたのはいいけれど，"S"で始まり"O"で終わるメーカーのを買ってきてしまったんです。シャープさんゴメンナサイ。しかし，私は間違いなくSuper MZを買ったのでありました。今から来年も優勝したら何を買おうかと2人で悩んでいます。
山本　雅昭（29）神奈川県

飛行機がおっこちたり，首都圏の国電が止まったり，たいへんな年でしたね'85年という年は。

◆つい先日，北陸へ修学旅行に行ってきました。兼六園のおだやかな緑や，能登金剛の荒々しい岩場，高山の古い町並みを目にして，何か心の中にしみじみとくるものがありました。しかし，高山の本屋さんでnamcoの本を買ってしまうあたり，コンピュータは1度染まってしまうと，なかなか忘れがたいもののようです。パソコンはこれからも趣味として続けて行こうと思っていますが，他のことにももっと目を向ければ，パッと何かが開けるかもしれないなあ，などと考えている今日この頃です。冬休みには鈍行列車で，どこかの古寺に行ってみたいと思っています。
溝口　健二（16）愛知県

◆ユーカラを使ってクラスの座席表を作成したら，先生方にたいへん好評を得ました。今までは，個人的なものにしか使っていなかったのですが，こんなに簡単なことでも他の人に役立つということはたいへん嬉しいことです。X1turboはやはり実務に使うことが狙いですからね。
山本　正幸（16）静岡県

他の人にパソコンを使う楽しみまでわかってもらうことがいちばん嬉しいですね。

◆ぼくのMZ-2000に16ビットボードを付けました。いまさら付けて何をするんだと言われそうですが，そんな声は無視して逆アセンブラを製作中です。完成したらOh!MZに投稿しようと思っています。もし16ビットボードを付けていて，これからマシン語を勉強したいと思っている人には，技術評論社から出版されている『98FAN BOOKS 2・98マシン語』をお薦めします。8088もなかなかおもしろいよ。　大村　則道（15）静岡県

◆ここ何カ月かの間，愛機のMZ-80Bが起きてきません。早く起こしてあげないと，永遠の眠りについてしまいそうで不安です。アア，ゲームがしたい，漢字を使いたい。80B，どうしてあなたは80Bなの？　片山　明秀（26）大阪府

◆X1turbo model30に，やっとのことで熱転写漢字プリンタを取り付けたゾ。さあこれからがんばろう。　酒井　弘志（33）宮城県

◆Oh!MZ 9月号のお見合いから2カ月，ついに私のだんな様MZ-2500がやってきました。"その筋"の弟（知る人ぞ知るドルーピー）が作ってくれたグラフィックツールでお絵描きしたり，FM音源で遊んだりしています。速いし，頭はいいし，値段は安いし，文句なしです……と，ついついのろけてしまう私です。　松岡　真理子（18）山形県

ウウッー！　2500が羨しい。

◆この前，あるパソコンショップに行ったときのことです。ある3人の子供の会話を聞いてしまっ

たのです。その会話の内容とは，A：「ROMってなんのこと？」　B：「俺もそれがわからなくて悩んでるねん」　C：「それは，このカートリッジのことやで」と言ってMSX用のROMカートリッジのゲームを指さしたんです。これは実話です。

田中　克彦（15）兵庫県

◆今度，マクドナルドでアルバイトを始めました。場所は山梨の甲府駅ビル店です。ぼくは身長170cmぐらいの顔のいい（本当だよ）少年ですし，ほかにもかわいい娘もいっぱいいますので，お近くにお出かけのときはぜひご来店ください。

前田　理（15）山梨県

◆Oh!MZを週間にという声をよく聞きますが，今の内容を週間誌にするにはチョット無理があります。そこで，Oh!MZminiという別冊はいかがでしょうか。A5判サイズで150ページ，200円。内容は，MZ関係はもちろん，他メーカーや海外の新製品情報，ハード，ソフトの売り上げBEST 10などで，これだけではOh!MZに関係ないから,MZ-5500/6500を使ったビジネスコーナー，シャープコンピュータ使用レポートなど。この内容で200円，さあどうだ。

串田　建二（17）新潟県

◆新年明けましておめでとうございます。昨年はS-OS誕生，MZ-2500発売と，MZ周辺は多忙でリッチでしたね。今年はOh!MZの読者はすべてマシン語に精通すべく，みんなでがんばりましょう。

得地　茂（29）北海道

旧年中はいろいろお世話になりました。今年もよろしくお願いいたします（この部分は1月1日以降にお読みください）。

桑原　誠（15）東京都

ぼくらの掲示板

仲間

★全国のMZ-2000/2200/2500のファンのみなさん，こちらは「水戸MZ 22クラブ」です。活動内容は主に情報交換を目的としていますが，特にフロッピーのソフトおよびハードに関して勉強しています。興味のある方はハガキか電話で連絡を。〒310　茨城県水戸市千波町1173-1　桃井秀明　☎0292(41)4787

★ぼくはMZ-700を持っていて，やっとプログラムも作れるようになりました。誰かぼくの情報交換の仲間になってくれませんか。ハガキを待ってます。〒920　石川県金沢市小立野3-4-52　多賀秀和（15）

★「X1LAND－えっくすわんらんど－」では，現在，会員を大募集しています。活動内容はロールプレイングやアドベンチャーなどのゲームの情報交換が中心で，毎月会報も発行しています。会費は1カ月100円，入会金は不要です。会員は現在約80名。詳しいことをお知りになりたい方は60円切手同封のうえ連絡を。〒679-53　兵庫県佐用郡佐用町佐用3124-1　新免一

★「MXC」では，現在会員を募集中です。毎月RPGやADVなどの特集を組んだ会報を発行しています。X1ユーザーの方ならどなたでも参加できますので，我と思われん方は60円切手を同封のうえ連絡を。〒960-06　福島県伊達郡保原町字竹内町2　水口昌郁

★X1ユーザーの皆さん「RUN X1京都」に入会しませんか。まだできたばかりのクラブですが，会報の発行や情報交換など密度の濃いクラブにしたいと思っています。興味のある方は60円切手同封のうえ連絡を。〒601-13　京都市伏見区醍醐中山町39-2市住中山団地17棟502号　黒沢健次

★「WHAT?! X1クラブ」では会員を募集します。X1を現在持っている人，またはこれから買おうと思っている人は，ぜひ当クラブに参加しませんか。入会費は無料です。詳しいことはハガキで連絡を。〒059-09　北海道白老町末広5-9-5　村岡通信

★X1ユーザーでOh!MZの読者の方，気楽に情報交換しましょう。まずは封書か往復ハガキで連絡を。〒190-11　東京都西多摩郡羽村町3-20-6　コーポ緑ケ丘101号　小川淳一

★現在，MZ-2500を購入して悪戦苦闘しているところです。誰か同じ機種のユーザーの方で，情報交換してくれる方を探しています。連絡はハガキで。〒780　高知県高知市青柳町44下知排水所内（南）　味元勝郎

売ります

★MZ-1500用プリンタMZ-1P14（ケーブル付）を3万円で。まずは往復ハガキで。〒959-24　新潟県北蒲原郡加治川村大字向中条844　中野道弘

★MZ-1500用プリンタGP-500Z＋ケーブルMZ-1C47を2万円で。〒859-32　長崎県佐世保市権常寺町876-2　原明博

★プリンタMZ-1P07（本体のみ）を2万5千円で。データミッターDR-100（トリオ）とMZ-2200用インタフェイス（ソフト付）を2台セット3万円で。近くの方を希望。ハガキまたはTELで。〒671-12　兵庫県姫路市網干区高田166-11　岩本光春　☎0792(73)9426（夜8時以降）

★漢字プリンタMP-80K（エプソン）定価18万9千円を5万円で。MZ-2000/2200で使用する方にはインタフェイス（I/Oデータ製）を付けます。またMZ-2000/2200用純正FDインタフェイスを1万5千円で。まずは往復ハガキで。〒011　秋田市飯島美砂町6-1-105　出雲正行

★X1用漢字プリンタCZ-80PKを3～4万円で。まずは往復ハガキで。〒010　秋田市川元むつみ町2-40　菊地金作

★X1用320KB外部メモリCZ-8EM＋64KBプリンタバッファを5万円で。〒546　大阪市東住吉区桑津1-9-24仲田荘　田窪伸晃

★X1用CZ-8RBとCZ-8FA，CZ-31FR（ソフト付）を半額で。まずは往復ハガキで。〒321-14　栃木県日光市相生町182　秋山吉康

買います

★スタティックRAMボードMZ-1R12を定価の半額ぐらいで。連絡はハガキで。〒567　大阪府茨木市平田2-22-16　前川新

★MZ-1500用の漢字ROM（MZ-1R23）＋辞書ROM（MZ-1R24）の両方セットで3～4万円で。またはどちらか一方だけでも可。連絡は希望価格明記のうえ封書で。〒037-03　青森県北津軽郡中里町大字豊島　松田幸喜

★データレコーダCZ-8RL1を1万2千円で。連絡は往復ハガキで。〒034-01　青森県十和田市大字深持字長下30　佐々木一夫

★X1C用拡張I/Oボックス＋漢字ROMを3万円以下で。連絡は往復ハガキで。〒316　福岡県大牟田市草木256　塚本雅彦

Oh!MZ バックナンバー

★1984年10月号を1,000円（送料別）で。切り抜きは不可。連絡はハガキで。〒519-21　三重県多気郡多気町相可231　逵浩也

★1984年9月号から'85年1月号までの5冊を各1,000円（送料込）で。落書き，汚れ可，切り抜きは不可。まずは往復ハガキで。〒636-03　奈良県磯城郡川西町唐院213　吉仲正和

★1984年8月号P.49～「超高速ペイントルーチン」と，'84年11月号P.127～「ハイパーペイントルーチン＆クロマキーペイントルーチン」の記事の切り抜き，またはコピーを両方あわせて500円以下（送料込）で。連絡はハガキで。〒189　東京都東村山市萩山町4-14-11　加藤隆

★1984年8月号を1,000円（送料別）で。切り抜き不可。もしくは当方のバックナンバー'84年3，4月号との交換も可。連絡はハガキで。〒852　長崎県長崎市音無町18-30　江里口裕之

★1984年3，4，5，8月号を各1,000円（送料込）。切り抜きは不可。まずは往復ハガキで。〒420　静岡県静岡市北1765-166　原田雅樹

★1983年11月号，'84年3，4，6月号の4冊まとめて5,000円，または各1,000円で。切り抜きは不可。連絡はハガキで。〒574　大阪府大東市灰塚4-336-23　西一夫

★創刊号から1983年4月号までをまとめて7,000円，または各1,000円（送料込）で。連絡は往復ハガキで。〒089-34　北海道中川郡本別町美里別東中　高木一明

◆掲載ご希望の方は，官製ハガキに項目（売る・求む……）を明記してお申し込みください。
◆ソフトの売買，交換については，いっさい掲載できません。
◆取り引きについては当編集室では責任を負いかねます。
◆応募者多数の場合，掲載できない場合もあります。

DRIVE ON

このコーナーは本誌年間モニタの方々より返ってきたレポートの中からご意見，ご希望，または気になる情報を抜粋して毎月皆さんにお届けしていきます。今月は11月号に対するモニタです。11月号の特集に対するご意見や，今月号の特集への要望などを中心にまとめてみました。

●あなたは建築デザイナー〈5〉は，レイアウトさえ変更してくれれば言うことなしです。図やイラストをうまく使い，フローチャートもあったので，全体の流れが手に取るようにわかることができました。プログラムも大きな文字で非常に見やすかったのですが，イラストはフリーハンドよりも，どちらかと言えば定規を使った線のほうがこの場合はよかったように思います。

田村　晴希（16）X1 熊本県

●11月号には，Z80命令表が掲載されていたので非常に助かりました。特集の記事内容は意外と簡単に理解することができたので，自分の基本知識がどの程度のものか判断することができたような気がします。その中でも山田氏の「オーナーズマニュアルが聖書です」は，これまでの自分の経験と同じような内容だったので共感を覚えました。ただ，特集のトビラに並んだタイトルを見ただけでは内容がサッパリわからないものがいくつかあったので，そういったものについてはサブタイトルを付けるなどの工夫が必要だと思います。また，本文中の専門用語などは，初心者にとって記事を読んでゆくうえでの障害となりかねないので，特集の最後や欄外にまとめて専門用語の解説文を付けてもらえば，もっとわかりやすいものになったのではないでしょうか。　紺谷　憲児（18）X1，MZ-731 大阪府

●Oh！MZ に掲載されているゲームプログラムについて気がついたことなのですが，読んでいると入力方法，遊び方，それではどうぞ楽しんでくださいとあったあとで，長い英数字のかたまりが続くのでは，せっかくのゲームの魅力が薄れてしまいます。せめて作者自身によるプログラムのPRや，編集室の推薦文やアドバイスをいっしょに載せてみてはどうでしょうか。

宮川　正雪（20）MZ-1500　東京都

●「霧降高原から／パソコンはバイクに乗って」は，読み物としてとても素敵だと思います。ただでさえ難解な専門用語の多いパソコン誌の，あのリストばかりの記事と，細かい文字ばかりを追っていると，どうしてもこんな読み物がほしくなってしまいます。このような記事はパソコン雑誌の清涼剤といったところでしょうか。ただ，ひとつだけ不満を言わせていただくとすれば，せめてあと1ページでいいですから，増ページをお願いしたいと思います。

松木　淳子（29）FM-New7，PC-6001 京都府

ごめんなさいのコーナー

12月号　ハード製作の基礎知識

p.34　右段26行目

セラコンの数字の意味は「これをabcとすれば容量はab×10^{C} pFです。たとえば103と書いてある場合，容量は10×10^{3} pF＝0.01μFです」の誤りです。

p.36　図14部品表

「8mmφスズメッキ線」は「0.8mmφスズメッキ線」，「カンハヤト」は「サンハヤト（基板）」の誤りです。

p.37　中段3行目

「LEDは足の短いほう，もしくは太いほう（つまりカソード）をICに向けて付けます」の誤りです。また，図16実体配線図左上の「⊣◄⊢」の記号は「⊣►⊢」の誤りです。

12月号　共通I/Oポートの製作

p.41　表2～5　部品リスト

基板としてサンハヤト「2000/2200用基板」，「X1用基板」ではよくわからないというお問い合わせをいただきました。型番はそれぞれMCC-157，MCC-153ですから，そのように注文すれば確実だと思います。また，コネクタの型番はFAP-26-03となっていますが，これには＃1～＃4の4種類があります。どれでも使用できますが，ハンダ付け用＃2がベストです。

p.44～47　図8～11

図中，セラコンの数字と容量が全体に不一致ですが，すべてパスコンですので103，104どちらを使用してもかまいません。

●MZ-80K/C/1200

図11-A部品配置図のボード内「注」の下のaはbの誤りです。

●MZ-700/1500，80B/2000/2200

図9-A 回路図のZ80A　PIO内$\overline{IORQ}$のピン番号26は36の誤りです。

図11-B，C実体配線図で，PIOの36番ピンとLS00の8番ピンをつなぐ線は，PIOの37番ピンと同LS00の8番ピンに変更して，さらにPIOの36番ピンとLS32の1番ピンを接続してください。

PIO(36)-LS00(8)→PIO(37)-LS00(8)
PIO(36)-LS32(1)

●X1

図10-B部品配置図で，PIOの右の2Kの抵抗は，右側のLS 32の右に置き，4番ピンの隣りに抵抗の足がくるようにします。また，3.3Kの抵抗が載っていませんが，これはLS 08の左側に置き，12番ピンの隣りに抵抗の足がくるようにします。実体配線図を参考にしてください。

図11-D実体配線図で，中央右の1Kの抵抗は3.3Kの誤まりです。また，左下LS00の1，2，3番ピンはこの順につないでショートさせたほうが動作が安定します。PIOのピンが左右1本ずつ足りませんが，10番ピンと30番ピンが足りないものと考えて，図中に追加しておいてください。

12月号　FM音源ボードの製作

p.53　ボードの写真（目次，31ページも同じ）左上にあるコンデンサは不要のものです。無視して製作してください。また，表4部品表の中で，O．P．の「4558×2」は「4558×3」の誤りです。なお，YM-2203，YM-3014が2セット手に入らない場合は，とりあえず1セットだけでも使用できます。その場合，ICソケットはすべて取り付け，右列，左列いずれか一方にYM-2203，YM-3014をさして使用してください。

12月号　感動のX1DXの製作である

p.59　表3　地獄の配線表

31，32番を次のように訂正してください。

－A7→IC3－2番（WRITE PROT）
－A8→IC3－4番（PACK）

なお，表のとおり配線してしまった場合も本体が壊れる心配はないはずです。

12月号　画面に愛を告白する方法

p.97　左段15行目

「MZ-2000のモニタでは11D1Hがそうで，11D1Hに X座標，11D2Hに Y座標……」の誤りです。

p.98　右段下から5行目

「A＝(A mod5)＋1」は「A＝((A＋1)mod5)＋1」の誤りです。

12月号　Prolog-85

「Prolog-85を走らせるには，85年7月号のS-OSのバージョンアップが必要です」との記述が落ちていました。これを実行していない場合はメモリオーバーのエラーが出ますので，S-OSの1F6AH番地からの2バイトを各機種次のように書き換えてください。

X1/X1turbo	……00FF
MZ-2000/2200(G-RAMあり)	……FFFF
MZ-80B/2000(G-RAMなし)	……00D0
MZ-700/1500	……00D0
MZ-80K/C/1200	……00C0

また，各機種次のアドレスの内容を00Hから01Hに書き換えておきましょう。

X1/X1turbo	……154DH
MZ-80B/2000/2200	……132AH
MZ-80K/C/1200/700/1500	……123AH

バグに関するお問い合わせは

☎03-263-2230

月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

S-OS大賞 清水和人賞 投稿大募集

▶シリーズ企画S-OSでは，共通化システム関係の投稿を下記の部門に分けて募集いたします。

1) システム賞：パッケージなどの追加によりS-OSのパワーアップに成功した例
2) アプリケーション賞：ユーティリティやツール，ゲームなどの開発例
3) 言語賞 別名こうもとやすひこ賞：各種言語（インタプリタ，コンパイラなど）の開発例
4) 発展賞：S-OSを他機種に移植して仲間を増大させる例

期限は特に設けません。優秀な作品を選んでそのつど誌上公開し，規定の原稿料および記念品を差し上げます。記念品はソニースポーツウォークマン(WM-F75)です。

▶6月号でお知らせした「清水和人賞」ですが，テキスト・アドベンチャー部門では，

北斗賞（スリル＆サスペンス）
南斗賞（ユーモア＆パロディ）

の2部門を設けました。アイデア，シナリオ，プログラムともに受け付けています。奮ってご応募ください。

なお，これまでのゲーム特別賞は「清水和人賞」としてテーマを限らず受け付けます。皆さんの力作をお待ちします。

〈賞品〉

清水和人賞 ―― ソニースポーツウォークマン WM-F75 32,000円
北斗賞・南斗賞 ―― カシオポケットテレビ 19,800円

▶今月のテーマはありません。ショート，その他の投稿も変わらずお受けしていますので，どしどしご応募ください。

採用になった作品には，本誌規定の原稿料をお支払いします。また，投稿作品のなかから特に優秀な作品には「月間特別賞」として原稿料とは別に記念品を差し上げています。今回の記念品はソニースポーツウォークマン(WM-F75)です。

皆さん，投稿をお待ちしています。

応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，プログラムは最低2回はセーブしてください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討の上，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒102 東京都千代田区四番町2-1
日本ソフトバンク出版部
Oh！MZ「㋢㋘㋮名」係

SHIFT BREAK

▶初登場のKYOです。先日，知る人ぞ知る駒場祭で，知る人ぞ知る河童踊りを踊って参りました。雨の中，日本酒をかっくらいながら，OBを交えて部員一同，踊り狂いました。最後にプールに飛び込むのですが，その水の冷たさは，とても筆舌には尽くせません。でも来年もやります。見に来てください。（KYO）

▶先日，国電が全面マヒした。私にはまったく関係なかったのであるが，友人の中にはその日に実験があったせいなのか，新幹線を利用してやって来る人もいた。ゲリラどもに怒っていると，今日は編集室が停電？ なぐり込みかと思いきや，ブレーカーが飛んだだけ。つかの間のサスペンス。（Y）

▶いよいよ年末，コンパが多くなる季節がやってきました。私の学校のクラブでも，いろいろと理由をこじつけては，コンパが開かれています。今コンパでは，かならず一気というのがありますが，だれがあんなくだらないことを考え出したのでしょう。酒はまずくなるし，死ぬ人も出てきます。やはり酒は好きなものを好きなだけのむのがいいことだと思います。（技）

▶下高井戸のとあるデパートの屋上に行ってみると，懐しいゲームがたくさん置いてありました。それもテレビゲームが出る以前のピンボールなどのたぐいのものばかりなのです。ご丁寧にも横でラムネまで売っていました。あまりの懐しさに眺めるだけで満足し，ゲームをやらずに帰りました。（こ）

▶この前，パソコンサンデーを見ていたら，白石さんがハレー彗星にかこつけて自分の誕生日を宣伝(?)していた。実はぼくも誕生日が同じなのだが，ぼくはそんなことはしません。11月27日だったなんて，恥ずかしくて口に出せません。ちなみにあのブルース・リーも同じ誕生日なんですよ。（Ku）

▶turboを買ってからちょうど1年がたちました。マニアタイプも使っていたので，もう3年もXIに付き合ってきたことになります。それなのに超初歩的なミスをしてしまいました。98用のプログラムを移植していて，入力がすべて終わりディスクのディレクトリを見たとき，思わず見慣れぬファイルネームのプログラムをLOADしてしまったのです。……なんて馬鹿なんだ！（K.A.）

▶明けましておめでとうございます。いや年じゃあありません。夜でんがなにぬねの。流石(るせきじゃないぞ！)にこの時季になるとこーゆー生活となり，私の健康はじわぁ，じわぁとムシバマレていくのです。コタツでぬくぬくとコンピュータ（略してコタコン）のし過ぎに注意しましょー。それしけても世間の，X'mas騒ぎがまるで別世界のよーであることよ。（IMT）

▶や，やっと解けました「デゼニランド」。その勢いで「ザ・キャッスル」，「WORRY」を終え，現在「ハイドライド」の城の所でもがいています。実はみんな途中でいやになって投げ出したものばかりで，再度挑戦しているというわけです。他にもあるんですよね「スターアーサー伝説Ｉ」とか……。（は）

▶引っ越ししてから早や2ヵ月。ようやく落ち着いてきたらこの寒さ。この前ヌクヌクと一日ストーブにあたっていたら，ムードが盛り上がってきて，ついに2年振りのハンダ付けとなりました。復帰第一作は RS-232C用のヌルモデム。PC-8201とMZ-2500の熱い対面となりました。やっぱり冬はハードがいいなァ。（M.K.）

▶K.S.さんへ。Willをやっていたら興奮してキーボードにビールを飲ませた話なんて書けません（恥ずかしい）。「イウ」がわからずカプセルの前で立ち往生したことも書けません。でも，堀ちえみのベストに「クレージーラブ」が入ってないのはケシカランということは書けるのです。（KO）

▶というわけで，問題はカツ丼である。私はその日カツ丼を食べる予定であった。それがダメになったとき私は動揺した。人は期待していたことが裏切られると寂しい。海がドッパーンで「海のバ□□□□カヤロー」と叫ぶのである。それが人生，それが男ぞ，とついつい4行目を19字で改行してしまう私であった。日本語は正しく使いましょう。（K.S.）

▶昔の映画の中では，すべてのコンピュータにはちらちら点滅するパネルが付いていたわけだ。そいでもって，今でも映画の中の化学の研究室にはフラスコの中に水とドライアイスを入れたものが置いてあるわけだ。そいで，最近はパソコンがけっこうＴＶドラマなんかに出てくるんだけど，未来の設定なのにCRTが320×198だったり，漢字が出なかったり，プリンタが異常に騒がしかったりするわけだ。やはり，一目で「あの安い機種だ」とわかるようなのはいけませんな。特に×××製のやつは。けけけ。（M）

▶先月の日曜日，東京は秋葉原という街の一角で，シャープ主催の「わんさかバザール」なるものをやっているというので，チョットだけのぞいてきました。会場はあまり広くなかったけれど，大勢のユーザーの熱気がムンムン。会場のあちこちを見まわして「あれもこれも安いなあ，みんな熱心だなあ」とただただ感心して帰ってきました。めでたし，めでたし。（N）

▶「FM音源サウンドエディタをチェックしたい」などと言って共通I/OポートとFM音源ボードを借りておきながら，ついつい遊んでしまった。いろんな音を選んでは鳴らしてみる。もっといい音にしようとデータをいじっても，なかなか期待どおりの音にはならない。FM音源って奥が深いね！（@）

▶A少年が「ボールブレイザー」(ATARI)を持ってきたため，その電光石火の3Dグラフィックスにド肝を抜かれ，いったい日本のパソコンは何をやってきたんだと怒りまくり，もはや68000 スーパーファミコン版の「マーブルマッドネス」でも移植して対抗するしかないと思う今日このごろ。えーと，あれ？あれ？（T）

microOdyssey

最近，新聞を読んでいると「身内のことが本になる『出版論』花盛り」という記事が目についた。その記事の内容は，なぜか出版界内部そのものをテーマにした本の刊行が相次いでいるというもので，その中に書いてあった「出版物の量的氾濫の中で，もはや書店は売れそうもない堅い本などは荷ほどきもせず，店内在庫本をトコロテンのように店外に突き出す」というくだりが気になったからだ。

私は，今でも頻繁に書店に足を運んでいるほうだが，ここ数年の街の小さな書店の店内の移り変わりを見ていると，どんどんと書籍を置いてあるスペースが姿を消し，雑誌やコミック，文庫本にその領域を占領されつつある実状が，歯止めの効かないままに現在まで続いているようだ。このままでは，昔のようにブラリと入った書店で気の向くままに本を選んで買って帰るといった，気軽な楽しみが奪われてしまうかもしれないし，このことは都会の書店を見ていてそう感じるのだから，地方の都市ではもっと切実な問題なのかもしれない。

学生の頃はよく，わからない言葉や数字があると図書館に行くのがおっくうで，近くの書店に飛び込んでは立ち読みで用を済ませたり，さらには，関係のない本までついでに読ませてもらったりと，本を買う以外の目的でずいぶんと利用させてもらったものだが，現在のように書店に並んでいなければならないはずの多種多様な情報源とも言える書籍類が，軽薄短小の時代の流れとともにしだいにその立場を弱められてしまうことは，たとえどんなに質の高いものであってもそれを知ってもらう場所を与えられないままに終わってしまう事にもなりかねないのではないだろうか。

何年か前，私がまだ学生のときに，ある文芸誌の編集者と偶然に酒の席で一緒になったことがあるが，そのときに酔った勢いで初対面の私に向かって，これからの文芸誌のありかたがどうしたこうしたと，延々と弁舌を振るわれて辟易としてしまったことがある。しかし，その中で一言だけ「自分達の仕事は，常に時代の波にもまれながら仕事をしているが，その波がひいたあとにどんなに小さなものでもいいから残すことができれば最高なんだけれど……」と言っていたのを今でも覚えている。

今になって考えてみると，その頃すでにそういった仕事の現場では，一種の危機感のようなものが存在していたのだろうが，その一部を最近になって新聞の紙面から思い起こすことになろうとは思ってもみなかった。もう読書の秋というにはあまりにも風が冷たくなってしまったが，そんなことでも思い出しながらこたつのなかに入って，偶然に見つけた一冊の本にゆっくりとつきあうゆとりを，これからも忘れないでいられればと思う。 (N)

'86. 2月号 1月18日(土)発売

- **特集 私が主役だ！**
  読者参加特別企画
- **全機種共通システム ディスク対応S-OS"Sword"**
- **BASIC DATA LIST Part.2**

■ バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(233)3312
	〃	書泉ブックマートB1	03(294)0011
	〃	書泉グランデ5F	03(295)0011
	八重洲	八重洲ブックセンター3F	03(281)1811
	新宿	紀伊國屋書店本店	03(354)0131
	渋谷	東急ハンズ寿楽洞7F	03(464)4604
	池袋	西武百貨店マイコン売場9F 西部ブックセンター11F	03(981)0111
	調布	真光書店	0424(87)2222
	町田	東急ハンズ寿楽洞	0427(28)2782
神奈川	横浜	有隣堂ルミネ店	045(453)0811
神奈川	横浜	横浜書店	045(241)5445
	藤沢	有隣堂藤沢店	0466(26)1411
	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	西武ブックセンター10F	0474(25)0111
大阪	都島区	駸々堂京橋店	06(353)2413
	北区	旭屋書店本店4F	06(313)1191
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
京都	中京区	オーム社書店	075(221)0280
長野	飯田	平安堂飯田店	0265(24)4545

定期購読のお知らせ

定期購読の申し込みをお受けしています。本誌が手に入りにくい地区にお住まいの方，毎月購読していただいている方，入手確実な定期購読への加入をお勧めします。詳しくは，本誌とじ込みの振替用紙をご覧ください。

バックナンバー在庫状況

'85 4, 5, 7, 8, 9, 10, 11, 12

以上の在庫がございます。

バックナンバーのご注文はお近くの書店からできますが，どうしても入手しにくい場合，直接弊社へ現金書留にてご注文ください。なお，郵送料は冊数によって異なりますので，前もってご連絡ください。お問い合わせは，出版営業（☎03-261-4095）宛お願いします。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS㈱にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区神田小川町3-5
☎03(291)2632

Oh!MZ 1月号

■1986年1月1日発行 定価480円 ■発行人 孫 正義 ■編集人 田鎖洋治郎

■発売元 (株)日本ソフトバンク

■出版部 〒102 東京都千代田区四番町2-1 ☎03(261)4095 FAX 03(262)8397

編集室☎03(265)5808

出版営業☎03(261)4095

広告営業☎03(255)9677

■本社 〒102 東京都千代田区九段南2-3-14 靖国九段南ビル ☎03(263)3690(代)

TELEX 東京 232-4614JSBTYJ FAX 03(263)3660(代)

■大阪支店 〒542 大阪市南区難波千日前5-19 河原センタービル3F

☎06(644)0191(代) FAX 06(644)0160

■印刷 凸版印刷株式会社

MZ-1P17
Printer Utility

 (5'FD)で

■カラーハードコピーができる(MZ-1P17対応)
● 〔BASIC〕で作成したグラフィック図形を(キー)を押すだけで、又〔嬉楽画〕で描いた絵でも(Utility使用で)簡単にカラーハードコピーができる。
● (キー)を押すだけで、テキスト画面とグラフィック画面を合成した、カラーハードコピーができる。
● カラーハードコピーの大きさを4段階に指定できる(キー操作)
● 作成した絵の色を4色(黄、紅、青、黒)に分解しそれぞれの色を抽出して、カラー印刷ができる。
● 上記の各特長を1つの〔BASICコマンド〕として、プログラムの中で使える。

■有名マイコンショップで、その美しさに感動して下さい!!

■必要システム構成
● 本　　　体:X1ターボ(5'FD)
● プ リ ン タ:MZ-1P17
● ケ ー ブ ル:MZ-1C48(X1/ターボ用)
● カラーリボン:MZ-6P17

6,000円

スーパーカラーBASIC

新しい発想

新しい感覚のBASIC

X1 turbo &C/CS/CK

■グラフィック図形を上下左右、自在に動かせる
● BASICコマンド〔MOVE〕を使って図形を動かせるのはもちろん重ね合せた色の中から指定色だけを動かせる事ができる。

■鮮明カラー36色高速ペイント
● φ~35の数値を入れるだけでコマンド〔PAINT@〕で中間色を含め36色の多彩な色が簡単にしかも超高速でペイントできる(テープ用及びノーマル5'FD用のみ)
● ターボ用はφ~71の数値を入れるだけでPAINTコマンドで72色の多彩な色が簡単に超高速でペイントできる。

■部分的にパレットができる
● 指定した範囲内の色を指定色に自由にパレットできる。(但し、本来のパレット機能より処理速度は遅い)

■グラフィック図形が反転できる
● 画面上の図形を上下、左右逆に表示することができる。
● 上下、左右対称の図形は、半分描けばコマンド〔REV〕で簡単に反転させて、一つの図形ができる。

■色を反転させることができる
● 指定した範囲内をコマンド〔RCOL〕で自由に色を反転できる。

■完全上位互換性で使い方が簡単
● テープ用は(CZ-8CBφ1)、ノーマル5'FD用は CZ-8FBφ1)、ターボ用は(CZ-8FBφ2)をサポートしていますので、コマンド、ステートメントはそのまま使える。

(詳細マニュアル及びサンプルプログラム付)

テープ用:6,000円
5 FD用:8,800円

MZ-1500

■グラフィック図形を上下左右、自在に動かせる
● BASICコマンド〔MOVE〕を使って図形を動かせるのはもちろん重ね合せた色の中から指定色だけ動かせる事もできる。

■鮮明カラー36色高速ペイント
● φ~35の数値を入れるだけで、コマンド〔PAINT〕で中間色を含め36色の多彩な色が簡単に表現できる。

■色を反転させることができる
● 指定した範囲内をコマンド〔REV〕で自由に色を反転できる。

■グラフィック図形を反転させることができる
● 画面上の図形を上下、左右逆に表示できる。
● 上下左右対称の図形は、半分描けば、コマンド〔REV〕で簡単に反転させて一つの図形ができる。

■一つのコマンド〔WINDOW〕で6機能
● 一つの図形を任意の位置へ同じ図形を表示できる。
● 異った図形の表示位置を交換することができる。
● 指定した範囲の図形を見えなくする。(マスク)
● 見えなくした範囲を解除(マスクの解除)し元の図形を見える様にする。
● その他

■完全上位互換性で使い方が簡単
● 標準BASIC(MZ-5Zφφ1)をサポートしていますのでコマンド、ステートメントはそのまま使える。

(詳細マニュアル、及びサンプルプログラム付)

QD用:6,000円

MZ-2000/2200

■グラフィク図形を上下左右、自在に動かせる
● BASICコマンド〔MOVE〕を使って図形を動かせるのはもちろん重ね合せた色の中から指定した色だけを動かせる。

■鮮明カラー36色高速ペイント
● φ~35の数値を入れるだけでコマンド〔PAINT〕で中間色を含め、36色の多彩な色が簡単に表現できる。

■色を反転させる事ができる(QD用のみ)
● 指定した範囲内をコマンド〔RCOL〕で自由に色を反転できる。

■4オクターブ3重和音
● 今までのMZ-2000系BASICになかったコマンドです。

■言葉が話せる
● コマンド〔TALK〕でローマ字読み入力で喋らせることができる。

■使い方が簡単
● 命令はすべてBASICですから初心者の方でも簡単に使える。
● テープ用はMZ-1Zφφ1を、QD用はMZ-5Zφφ7の標準BASICをサポートしていますので、完全上位互換性を保っています。

(詳細マニュアル(テープ用は説明書)、及びサンプルプログラム付)
2000用は要G-RAM I.II.III.

テープ用:4,000円
Q D 用:6,000円

通信販売のお知らせ

● 機種名及びテープ、QD、5FDかを明記し、住所、氏名、TELを記入の上現金書留(送料サービス)にて送付して下さい。

〒546 大阪市東住吉区湯里1-1-1 稲田ビル403号
マイコンシステム企画
電話 大阪 06(704)9923

SHARP X1 turbo MZ-2500用
UK-Turbo 財務管理システム
漢字版
新発売
その日の処理は
その日のうちに！
価格
¥48,000
入金伝票
出金伝票
振替伝票
●伝票を仕訳日記帳へ記帳する要領にて入力
●入力はコード入力で早くて簡単
●データの訂正・追加・削除OK！
●登録科目数160科目
●800仕訳／月の処理
●CRT上にて全財務諸表表示
仕訳日記帳
総勘定元帳預金
総勘定元帳売上
総勘定元帳仕入
現金出納帳
(株)ヒサゴ製
元帳用紙No.241使用
合計残高試算表
合計
合計金額10億円未満
科目コード一覧表
最高160科目
漢字コード一覧表
最高160個
経費一覧表
貸借対照表
損益計算書
★入力終了後即各財務諸表作製可能
★プリンターはシステムでサポートしているものを使用
(MZ-1P10A用は高速高品位印字)
★MZ-1P11A、UP-130K用は罫線付高速高品位印字を55,000円にて別に用意しています。
天草パソコンセンター
ウラカワ電器店
熊本県本渡市亀場町亀川1693－1
電話　0969－23－2813
お求めは
お近くのパソコンショップ又は
シャープ製品取扱店へ
※直販の場合は送料サービス
UK-Turbo
資料請求券
Oh! MZ
86.1

ズバリ!お買得特集

16ビットボード付 MZ-2200 パーソナルコンピュータ

MZ-1M01+MZ-1T02+MZ-LOGO
合計236,600円が
たったの
65,000円!

(機能アップ!!)
- 膨大なアドレス空間を確保。
- 処理能力が飛躍的にアップ!
- 豊富なソフトをそのままサポート。
- 16×16ドットの見やすい漢字表示。

16ビットパーソナルコンピュータ MZ-5500

本体(MZ-5521)¥388,000を
大特価¥148,000
本体(MZ-5511)¥288,000を
大特価¥109,000

MZ-5500シリーズ周辺機器
- 拡張ボード(MZ-1U05)………¥12,000⇨大特価¥10,000
- 増設ビデオRAM(MZ-1R09)…¥35,000⇨大特価¥25,000
- 漢字ROM(MZ-1R10)…………¥30,000⇨大特価¥18,000
- 増設RAM(MZ-1R11)…………¥80,000⇨大特価¥40,000
- 辞書ROM(MZ-1R14)…………¥40,000⇨大特価¥26,000
- 増設RAM(MZ-1R16)…………¥30,000⇨大特価¥26,000
- 日本語ワープロ(MZ-2Z025)…¥49,000⇨大特価¥26,000
- 統合化ソフトToday(MZ2Z014)・¥68,000⇨大特価¥35,000

最少の投資で、最大の価値を。

パソコン、周辺機器、ソフトなどプロの目で選んだ一級品ばかりを限界価格でお届けします。
マイコンライフをよりグレードアップしたい、と願っているあなたにとって耳よりな情報ばかりを満載!
本誌発売時には、下記価格表より、さらにお求めやすい価格に変更されている場合があります。

新製品!Super MZ
- シャープMZ-2500シリーズ

Model 30(MZ-2521)…¥198,000⇨

X1turboII(CZ-856C)¥178,000⇨

年末大特価

MZ-1500……………… ¥39,800

- シャープCZ-850C……………¥168,000⇨¥75,000

X1シリーズ年末謝恩特価
例えば、セットの場合
- X1Fモデル10+14インチカラーモニター(RGBケーブル付)
  …………………………¥154,600⇨¥99,800

その他、X1モデル10、20。X1turboモデル10、20、30、40
年末謝恩特価販売!

- NEC PC-9801E……………¥215,000⇨¥149,000

モニター
- シャープMZ-1D10(4050文字)‥¥41,800⇨¥28,000
- シャープCZ-801D(カラー)……¥99,800⇨¥59,800
  (640ドット・チューナー付、14インチ)
- PC-8047(アンバーイエロー)……¥46,800⇨¥18,000
- NEC PC-8052(カラー)………¥118,000⇨¥39,800
  (ドットピッチ0.39m/m、RGBケーブル付)14インチ
- NEC PC-TV151(カラー)………¥94,800⇨¥53,000
  (510ドット、RGBケーブル付)15インチ
- NEC PC-KD552K(カラー)……¥112,000⇨¥59,800
  (ドットピッチ0.39m/m)14インチ
- NEC PC-KD551K(カラー)……¥99,800⇨¥69,800
  (640ドット)
- シャープCZ-150DS(カラーTV)‥¥98,000⇨¥54,000
  (640ドット、RGBケーブル付)15インチ
- シャープ20-202C(カラー)……¥168,000⇨¥48,000
  (640ドット、RGBケーブル付)20インチ
- 東芝14V20F<W>(カラー)……¥99,800⇨¥54,000
  (RGBケーブル付)14インチ
- ナショナルTR120M1C(グリーン)‥¥46,800⇨¥15,000
  (2000文字、ケーブル付)12インチ
- シャープ2000文字Cu-14F1……¥64,800⇨¥39,800
- シャープ4050文字Cu-14H2……¥99,800⇨大特価!
- シャープ4050文字Cu-14A1…¥128,000⇨大特価!
  (アナログ・RGB対応PC-9800シリーズ対応)
- シャープ15M-412C(カラー)……¥118,000⇨¥44,800
  (640ドット、RGBケーブル、チルト台付)

プリンター
- シャープCZ-81P(X1用カラープロッタープリンター)
  ………………………………¥34,800⇨¥13,800
- NEC NM9400-(136桁)……¥310,000⇨¥165,000
- NEC漢字PC-PR104…………¥148,000⇨¥92,000
- NEC PC-6022…………………¥39,800⇨¥9,800
- シャープMZ-1P09………………¥47,600⇨¥30,000
  (1500用ケーブル付)
- シャープカラープリンタMZ-1P17・¥79,800⇨¥67,800
- シャープCZ800P(X1用)……¥142,800⇨¥49,800
- シャープMZ-1P07………………¥95,000⇨¥79,500 (ケーブル付)
- シャープMZ-1P01(MZ-700用)‥¥39,800⇨¥15,000 (インターフェース・ケーブル付)
- CZ8PD2ドットプリンター…………¥79,800⇨¥49,800
- 日立MP-1041…………………¥169,800⇨¥85,000
- シャープMZ-80P4B(136桁)………ズバリ¥79,500

拡張機器他
- PC-6601-01(拡張漢字ROM)…¥24,800⇨¥14,800
- シャープ拡張I/Oボード
  MZ-1U01(MZ2000用)…………¥37,000⇨¥27,800
- MZ-2200用キーボードのみ………………………¥12,000
- シャープMZ8BG…………………¥39,000⇨¥19,800
- MZ-8GBK(MZ80B用)…………¥39,000⇨¥22,000
- シャープMZ8BGK………………¥19,800⇨¥18,000
- シャープMZ-80I/O(80Kシリーズ用)・¥29,500⇨¥15,000
- シャープ漢字ROMボードMZ-1R13……¥41,800⇨¥35,500
- グラフィックRAM MZ-1R02×2…¥16,000⇨¥11,200
- シャープMZ-1R01+1R02×2……¥55,000⇨¥28,000
- CZ8KR(X1用)…………………¥32,000⇨¥17,800
- CZ-8VC(X1用)…………………¥15,800⇨¥13,400
- パソピアPA-7245(32K)…………¥40,000⇨¥28,000
- 日立MP-1802A(インターフェース)……¥15,000⇨¥7,500
- MZ-2000シリーズ232Cカード……¥19,800⇨¥16,800
- FM8用232Cカード………………¥30,000⇨¥24,000
- パワースプライザー(各種パソコン/フロッピーに使用可)
  (12V0.1Ah・12V2Ah・5V2.5Ah)…………………¥1,500
- シャープCZ-81EB(X1用拡張ボックス)
  ……………………………………¥29,800⇨¥14,800

フロッピーディスク
(インターフェース・ベーシック付)

シャープCZ-800F(在庫限)……¥79,800

日立MP3560…………………¥75,000

- シャープMZ-1F07……………¥158,000⇨¥98,000
- シャープCZ-501F(X1用)……¥129,800⇨¥105,000
- NEC PC-9831-4W……………¥198,000⇨¥138,000
- NEC PC-6601FD1………………¥39,800⇨¥25,000
- ティアックFD55B(増設用)……………………¥28,000

その他
- MZ-2Z004(F DOS)……………¥50,000⇨¥42,500
- MZ-LOGO……………………¥9,800⇨¥4,500
- X1NEW BASIC111SF……………¥8,800⇨¥7,500
- ニデコ・カラーボードNH-MZD2(MZ80K/C用)
  ……………………………¥69,800⇨¥10,000.!!

16ビットボードキット
MZ-1M01+漢字ROM………¥26,000

全国通信販売
北海道から沖縄まで
創業26年の信用と実績が安心をお届けします。

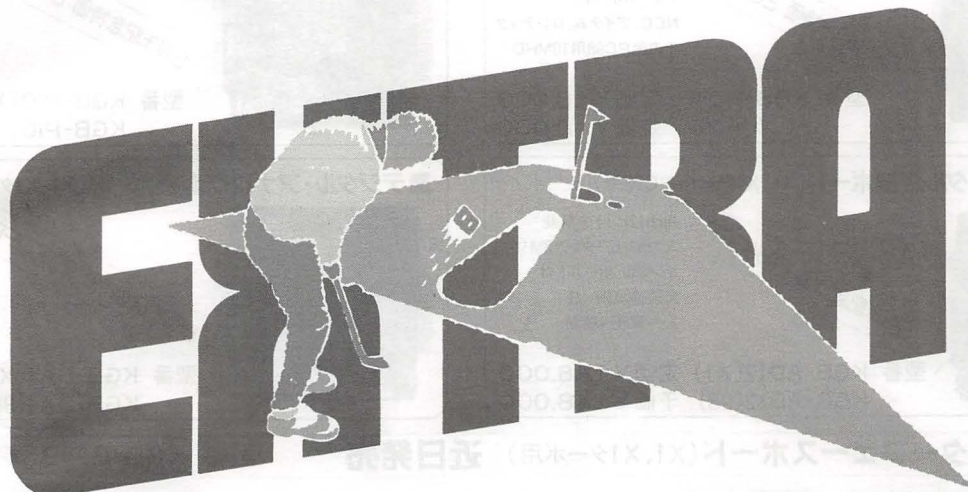
★送料はご注文の際にお問い合わせ下さい。
★当店では中古品は扱っておりません。
★ご注文は在庫を確認の上、現金書留または銀行振込でお申込下さい。全商品、クレジットでも扱っております。
★お申込みの際は必ず電話番号を明記して下さい。

アイビット電子(株)
営業所:〒192東京都八王子市北野町560-5
☎0426-45-3001〜3
☎03-545-0022 FAX.0426-44-6002

- 営業時間:10:00〜19:00
- 電話受話:20:00迄可
- 定休日:日曜日

■本誌の前号、前々号の広告もご参照ください。

EXTRA HYPER

第二世代へ進化

EXTRA HYPER

テープ版のIPL・ソフトをワンタッチでディスクに引き上げる
HYPER DISK MONITORが、
ついにEXTRA HYPERに進化しました。
専用のDATA DISKにテープ版ソフトを収容するので、
2枚、3枚と追加使用するとあなたのライブラリーがどんどん広がります。
EXTRA HYPERで扱えるソフトの種類はなんと125にもなりました。
もちろんシングル・ドライブでも使えます。

X1(要G-RAM)・X1turbo用

EXTRA HYPER (DATA DISK 1枚付)

5¼″FD・3″FD 各¥10,000

お近くのマイコン・ショップでお求め、又はご注文ください。詳しい内容をお知りになりたい方は営業部までお電話ください。（AM9：30〜PM5：30。日曜、祭日は除く）
当社直接の場合は営業部へ現金書留か郵便振替（東京6－123648株式会社ブルー・スカイ）又は銀行振込み（第一勧業銀行自由が丘支店普通1099629）でお願いします。
振替や振込みの場合は、住所・氏名・電話番号・商品名・機種名・メディア名をハガキでお知らせください。商品送料は不用です。

BLUE SKY Co.

株式会社 BLUE SKY 本社 〒411 静岡県三島市加茂16-4 営業部 〒152 東京都目黒区緑が丘2-17-17 電話 03-724-7980

MZ-2000シリーズ

H.Sコントローラー(テープ版IPL・ソフトがディスクに入る)5¼″FD ¥9,600
修理屋さん(ディスクなどに直接機械語が書ける) 5¼″FD ¥4,600
H.S4200(1枚のディスクを12分でテープにバック・アップ)5¼″FD ¥7,400

X1シリーズ

漢字 DEVICE MONITOR for TURBO
(ディスクなどに直接機械語が書ける。ターボ専用) 5¼″FD ¥5,600
修理屋さん(ディスクなどに直接機械語が書ける) 5¼″FD・3″FD 各¥4,600

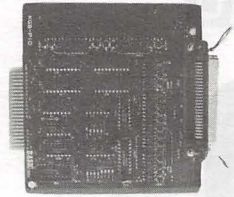
資料請求券 Oh!MZ 1月

信用と実績を誇る **BASIC HOUSE** 宇都宮の本格的な

マイコンショップ

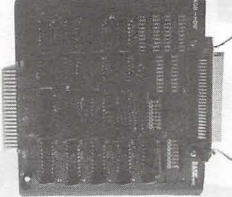
FA-LA

X1 X1 turbo シリーズ　**本格的各種インターフェースボード　発売中**

■ハードディスクインターフェースボード（X1ターボ用）

5周年記念特価 20%OFF

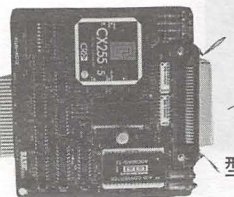
X1ターボで10MBのハードディスクを使用するインターフェースボード
NEC、アイテム、ロジテック
その他PC98用10MHD

型番 KGB-HDIF　定価￥18,000
ケーブル　定価￥ 8,000

■絶縁型パラレル入出力ボード（X1、X1ターボ、PC98用）

5周年記念特価 20%OFF

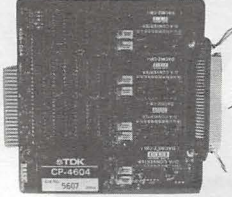
入力数 8入力 2ポート
出力数 8出力 2ポート
入出力フォトアイソレーション
入力電圧 5V〜18V
出力 オープンコレクター

型番 KGB-PIO(X1)　定価￥52,000
KGB-PIO(98)　予価￥52,000

■アナログ：デジタル変換ボード（X1、X1ターボ、PC98用）

5周年記念特価 20%OFF

16ch12ビット分解能
入力インピーダンス2MΩ
サンプル/ホールド付
変換速度25μS
入力電圧 4種類

型番 KGB-AD12(X1)　定価￥148,000
KGB-AD12(98)　予価￥148,000

■デジタル・アナログ変換ボード（X1、X1ターボ、PC98用）

5周年記念特価 20%OFF

4ch12Bit分解能
電圧出力 ±10V（標準）
ラッチ回路付

型番 KGB-DA4(X1)　定価￥128,000
KGB-DA4(98)　予価￥128,000

■GP-IBインターフェースボード（X1、X1ターボ用） 近日発売

型番 KGB-GPIB(X1)　予価￥68,000

パソコンで計測制御をしょう

PC・MZともテスト用プログラム、回路図、説明書付

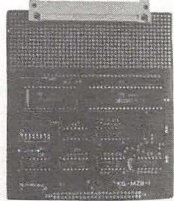
超低価格でホビーから本格応用まで可能!!

5周年記念特価 20%OFF

PC-8001 ※
PC-8001mkII
PC-8801
PC-8801mkII

型番 KGB-PC1
定価￥19,800

※専用のI/O BOXが必要です

MZ-700 ※
MZ-1500
MZ-80B
MZ-2000
MZ-2200

型番 KGB-MZ1
定価￥19,800

貴殿の考えているシステムが可能かどうか無料でコンサルティングします。

BASIC HOUSE SOFT

5周年記念全ソフト 20%OFF

PC-9801用アプリケーションソフト

PC-SEET（PC-PAL日本語版）

新発売　型番B9-6501　定価￥30,000

本ソフトはPC-PAL日本語のデータプリントアウトユーティリティソフトで最大印字桁数が310文字まで縮少文字を印字する事ができます。

※PC-PAL日本語は㈱大塚商会の製品です。

世界初!! 驚異の大ヒット システムソフトウェアコンバータ

BASIC HOUSE

パソコンテレビX1がソフトだけでMZ-2000 MZ-2200 PC-8001に早変り!!

品名	型番・内容	機種	定価
システムソフトウェアコンバータ 第1弾!!	B6-2213 MZ-2000 BASIC	機種：X1、X1C、X1ターボ	定価￥3.800
システムソフトウェアコンバータ 第2弾!!	B6-2217 LOGO and PASCAL	機種：X1、X1C	定価￥4.200
システムソフトウェアコンバータ 第3弾!!	B6-2218 システムプログラム andマシンランゲージ	機種：X1、X1C	定価￥4.200
システムソフトウェアコンバータ特別企画!!	B6-2220 N-BASIC	機種：X1、X1C	定価￥4.800
X1 ディスアセンブラ	B6-2109 Z80逆アセンブラ	機種：X1、X1C、X1D	定価￥4.200
MZ-1500ディスアセンブラ	B4-2101 Z80逆アセンブラQD版	機種：MZ-1500	定価￥4.800

各種BASICテキストコンバータ　絶賛発売中！

機種	元	変換先	型番	価格
X1	PC-8001	→CZ-800	B6-1483	￥3,800
	PC-8801	→CZ-800	B6-1493	￥3,800
	MZ-80B・2000	→CZ-800	B6-1413	￥3,800
	MZ・80K C・1200	→CZ-800	B6-1433	￥3,800
	PC-6001	→CZ-800	B6-1473	￥3,800
MZ-700	PC-8001	・MZ-700	B5-1483	￥3,800
	PC-8801	・MZ-700	B5-1493	￥3,800
	PC-6001	・MZ-700	B5-1473	￥3,800

定価￥3.800

MZ-700はじまって以来の超大作ADVENTURE GAME

画面数120枚の超大作 アップル風本格的アドベンチャーゲーム

ヒント集あります（送料￥100）

AUTOCRACY

絶賛発売中！　定価￥3,800

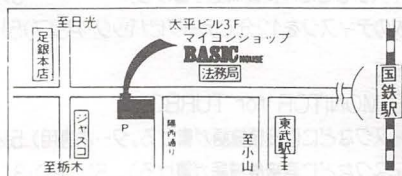
X1 turboで本格的なデータ通信をどうぞ!!

TURBO NET 5インチ2D

TURBO NET通信ソフトは沖電気、日立、NEC、田村電機、シャープのインテリジェントテレホンをサポートしております。カプラーもOK。

定価￥12,800

おわび

システムソフトウェアコンバータ(26-2213)がX1 turboで動作せず、ご迷惑をおかけしましたがX1 turboでも使用できるようになりました。

◆当社の製品はお近くのマイコンショップ等でお求め下さい。また、製品等に関するお問い合わせは直接当社へお願いいたします。

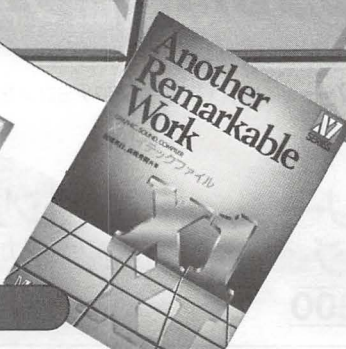
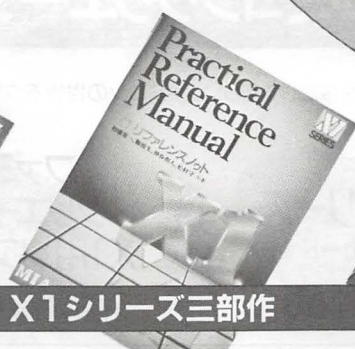
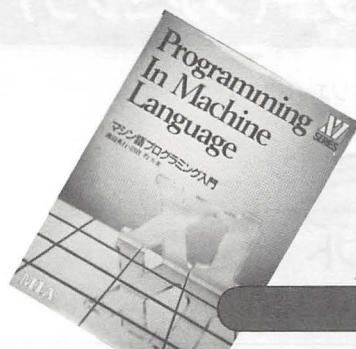
送料ソフト￥200　ハード￥500

マイコンショップ BASIC HOUSE

〒320 宇都宮市桜3丁目2−17 太平ビル
☎0286−33−1994　3F（株）計測技研
FAX 0286-34-1264　4F AD CORPRATION

X1 究極のメニュー

- ●X1　マシン語プログラミング入門
- ●X1　リファレンスノウト
- ●X1　ハイテックファイル

マシン語からゲーム作成ノウハウまで
基礎知識から実践的活用法まで
X1の総てを網羅したMIAの3大メニュー。
X1シリーズ3部作をどうぞ御賞味ください。

X1シリーズ三部作

X1マシン語 プログラミング入門

渡辺英行・沼倉均共著
A5判　定価2,200円（〒250円）

本書は、X1でマシン語を学ぶために必要な基礎知識から、実践テクニックまでを解説したマシン語入門書です。IOCSやI/Oポート、マシン語命令表などは、資料として見やすいように整理しました。既にX1でマシン語を使っている方にも十分活用していただけます。また、マシン語プログラムの開発用に、エディタ・アセンブラのリストも公開。

X1 リファレンスノウト

杉浦勇一他著
A5判　定価2,500円（〒250円）

X1のすべての機能解説と、その活用のノウハウを一冊に盛り込んだテクニカル資料集。X1の一歩進んだ実践的活用法を学びたいというユーザーに、ぜひおすすめします。X1 turbo にも対応！
内容：Hu BASICの内部構造／画面構成／周辺I/O デバイス／PSG／カセット／フロッピーディスク／IPL ROM／etc.

X1 ハイテックファイル

渡辺英行・高橋秀樹共著
A5判　定価2,200円（〒250円）

これまで最も希望の多かった項目を一冊にまとめました。内容はゲーム作成のノウハウ。LINEやPAINT などのグラフィックス。音声合成、エコーなどの効果音。開発ツールとして小型高速コンパイラ"Stellar"の全リストと使い方など、一味違います。特にアルゴリズムが重要な部分はBASIC で記述。X1ユーザーに贈る最後のメッセージです。

ROUND SYSTEM LABORATORY INC.

Super mz MZ-2500 『スーパー財務/テレビ元帳』¥128,000

MZ-2500の大容量・超高速をフルに活用した16ビット用ソフトを遥かに超える高速多機能会計ソフトの誕生です。

SUPER MZには「スーパー財務/テレビ元帳」今これに追随出来るものはありません!

①1枚のディスクに6,000仕訳のデータが入り、これを1ケ月分としても、12ケ月分としても使用出来る。
②勘定科目は補助科目を含めて600まで、全部自由設定。期中に追加、変更、割込が自由に出来る。
③指定期間内であれば、以前の月でも来月でも仕訳データの入力、訂正、削除が出来て、処理時間なし。
④仕訳データは日付順に入れなくても、仕訳日記帳も、元帳も、日付順、入力番号順の両方出せて、待時間なし。
⑤データ入力直後に電源が切れても、データの異常は起らない。誤入力のためデータが乱れても修復出来る。
⑥他のソフトの様にデータ量が多くなると処理時間が数分～数十分かること一切なし。いつでもすぐに出る。
⑦摘要の漢字入力は辞書ROMで文節変換、人名、地名も可。英数字、カナ入力も出来る上、パスワード入力は結合可能。(パスワードプラス機能)
⑧階層メニュー方式、オールメッセージ、誤入力時の警告、コマンドの常時表示のため初めての人でも殆んど説明書不要。
⑨サンプルデータ付のため、入手後すぐ全機能のテスト、プリント、データ入力の練習も出来る。

「スーパー財務/テレビ元帳」は今やあらゆるコンピュータ用会計ソフト中最高位のもので、これより高価なものでも、とうてい及びません。これはSUPER MZの優秀性とラウンドシステムの会計処理のキャリアの相乗効果とも云えます。経理事務の実務上のことを十分に配慮してありますから、実務家各位には十分ご満足頂ける内容です。但し全く簿記も分らない方は、その方の勉強を一寸だけお願いします。(全自動仕訳と云うものは世界中にありません。)

項目	内容
適合機種	あらゆる業種、法人、個人、特殊法人、組合、団体
勘定科目	全部自由設定、簡易科目名漢字入力、カナ漢字変換
補助科目	任意の科目に任意の数の補助科目設定可
勘定科目数	補助科目を含めて600個まで
仕訳件数	1枚のディスクに6,000件、最大12ケ月分に自動配分
金額	1件、合計共99億円まで。(オプション999億円)
摘要	漢字12字、カナ24字、パスワードプラス機能
マスターファイル	自動月次残高算出機能付ランダムファイル
データファイル	超高速日付順検索付ランダムファイル
使用言語	SUPER BASIC＋機械語
演算速度	16ビット機用ソフトの2倍強(当社比)
プリンタースピード	プリンターの限界速度で連続ノンストップ
プリンター用紙	全部普通のストックフォーム、元帳は専用用紙もあり

項目	内容
画面出力	テレビ元帳、テレビ試算表、テレビB/S,P/L、テレビ仕訳日記、テレビ予算実績対比、テレビ資金繰実績、当月、通期利益表
印刷出力	総勘定元帳、補助簿、試算表、貸借対照表、損益計算書、仕訳日記帳、資金繰実績表、予算実績対比表、その他
オプションソフト	特殊法人決算書、部門別利益計算書、工事台帳、手形管理、固定資産台帳(予定)
機器構成	MZ-2500 FD×2、128KB増設RAM MZ-1D22(CRT)又は同等品 MZ-1P18(漢字プリンター)又は1P10A,1P11A,1P06,1P03、辞書ROM
提供メディア	3.5インチ2DDフロッピーディスク×2
附属品	サンプルデータ、予備ソフト、ガイドブック

スーパーシリーズビジネスソフトは、「スーパー給与」「スーパー販売/テレビ台帳」「スーパー仕入/テレビ台帳」等続々発表の予定です。また熱心な自作派ビジネスマンのためにノウハウ公開の新Qシリーズはオールランダムファイルで発表の予定です。またMZ-80B, MZ-2000,2200用の「スーパー財務/テレビ元帳」(カナ)や「スーパー在庫管理」(カナ)やQシリーズ、テープソフトなど引続きサポート中です。詳しくは「SHARP MZ APPLICATION」をごらん下さい。尚、当社は誇大広告や欠点を隠して売ると云ったことは一切ありません。不愉快な電話をおことわりします。業者取引はメーカー認定店に限ります。
資料のご請求は、ソフトの種類を具体的に指定の上、なるべく切手200円同封して下さい。

総合カタログMZ版(No.3)〒200(No.4は61年発行)
★ユーザー直接のご注文を歓迎します(会員特典あり)
Dシリーズソフトのユーザーはスーパーシリーズは特別価格
〈ご注意〉当社ソフトのレンタル、コピイ販売、用紙の複製、商標の無断使用はバチが当たります。

※ご注意:テレビ元帳は当社の創作語で商標登録申請済です。(無断使用に重ねて警告します)

▶LODE RUNNERは米BRODER BUND社の登録商標です。
おもしろさ大接近!!
ワクワク!!
ワクワク!!
キミたち、人気のロードランナーに、
この秋、ひとつ仲間が登場!その名も
「チャンピオンシップ・ロードランナー」
ゴキゲン、痛快無比!
楽しさ120%のオモシロSOFT。
Championship
Lode Runner™
MZ-1500 QD版
チャンピオンシップ
ロードランナー
発売開始!!
•おまたせしました•
定価5,000円
ロードランナー
MZ-1500 QD版
定価5,200円
[お知らせ]
ロードランナーで遊びすぎたり、QDに傷が付いたり、その他の事情で画面データ(B面)が、うまく読み込みができなくなった人のために、画面データ(B面)を作成するプログラム(メンテナンスQD)をお送りします。新しいQDにコピーして使うと 何度でも使えます。QD代金、送料共800円分の切手同封のうえ、左記メンテナンスQD係宛送ってください。
(注:A面のプログラムは入っていません)
SUPER SOFT WARE LAB.
UNiVERSE
〒700 岡山市下中野519-1 TEL(0862)44-1176 [年中無休]PM1:00～PM7:00
●通信販売ご希望の方は現金書留にて上記ユニバース宛ご注文ください。
(送料無料サービス)

《《　漢字に仲間が増えた　》》

ワープロ NEW mini

MZ-80B・MZ-2000/2200・X1・X1 turbo
－－－－－－－－－－－－－ 価格 59,800 円
ワープロminiをお持ちの方は差額にて交換

［簡単な漢字変換］

これまでのかな漢字変換は、１文字単位又は、熟語単位に入力を行ない、そのつど変換する方法でした。これに対して NEWミニは、文節変換や一括変換ができます。一括変換は、文書すべてをかなで入力し、その後漢字に変換したい所にカーソル移動させて［CR］を押すだけで次々とかな漢字混じり文に変換するというものです。これによりキー操作が簡略かされて、スピーディな文書作成を実現し、文書作成中の思考の断続がないために、頭に浮かんだ通りに、気軽に文書が作れます。
一度漢字に変換した後に、漢字位置で［CR］を押すだけでひらがなに戻ります。

かんじをかんたんににゅうりょくする。
↓
■んじをかんたんににゅうりょくする。
↓
漢字を■んたんににゅうりょくする。
↓
漢字を簡単に■ゅうりょくする。
↓
漢字を簡単に入力する。

［便利な部首検索］

漢字の読み方がわからない時に、辞書から部首や画数で探すようにシステム内に 200の部首ファイルを持っており、目的の部首を指定した後、対応する漢字が表示され、その中から漢字１文字を探すことができ、難読文字の呼び出しが簡単になりました。ＪＩＳ第一水準、第二水準文字完全対応しております。

一丨丶ノ乙亅二亠人儿
↓
亰亥亶亨享京亳交商充

［豊富な辞書］

登録済の漢字は、30,000語。日常よく使われる漢字を厳選してあります。さらに、ユーザーが10,000語追加登録することが出来文書作成中から簡単に追加・閲覧することもできます。自動学習機能により最終使用文字が優先して表示され、使えば使うほど変換効率がアップします。単語は、6,000語の音訓登録されていて文節変換や一括変換をいっそうスムーズにしています。

［文字種類］

第一水準2965文字・非漢字・第二水準3384文字・外字80文字が使え、全角・倍角・半角・上つき・下つき（16文字）で表示することが出来ます。第二水準文字は、システム内に内蔵されていますので、大変便利に使えます。MZ-1P07 の様なドットプリンタでも第二水準の文字を印字することができ24ドットプリンタは、第二水準ＲＯＭが必要です。

［高性能な計算機能］

高性能な15桁計算することができ、電卓計算は、文書作成中にコンピュータのそばに電卓があるつもりで使えます。また画面計算は、文書内の数字を読み取り計算数値として扱えます。多彩な答え代入機能があり計算によって出た答えを全角や半角文字で文書内に代入できたり、右付けや金額などの自動コンマ付けすることも出来ます。
この計算機能で、単に漢字かな混じりの文書作成機としてだけでなく広義の文書作業を達成することができます。

［多彩な印刷機能］

ドットプリンタと24ドット漢字プリンタの機能100%活用し、書式指定で指定された、横書き・縦書きで印刷され文字間隔・行間隔を１ドット単位にコントロールできます。枚数指定や、自動頁付け、部分印刷などがあり、各種の案内で宛名だけを変えて何枚も印刷することができる差し込み印刷や文書ファイルから指定された文書から文書までを指定間隔に再生しながら指定枚数印刷できます。
MZ-1P10/11は、縦倍文字は印字不可能。

［ＰＯＰ機能］

X1用ではスーパーインポーズ機能を使ってワープロで作成した文章をバックの色、濃度、文字の色、スクロールスピードなどを指定して、店頭や待合室、ロービーなどでテレビ画面に表示することができます。

［ビデオ編集機能］

X1用ではデジタルテロップ機能を使ってワープロで作成した文章をバックの色、濃度、文字の色、表示行数（1,2,ブロック表示）などを指定して、ビデオテープにタイトルや各画面のコメントをビデオ録画できビデオ教材、ビデオレポート、ビデオPOPが簡単に作成できます。

［　機器構成　］

本　体	品　番	漢字ROM	グラフィック	プリンタ	
MZ-200.2200 MZ-80B	V6.1DG V5.1DG	MZ-1R13 MZ-1R13 (PIO-3055)	MZ-1R01+02 MZ-80BG	MZ-80BP5 MZ-80P6 MZ-1P07	EPSON MZ-1P10 MZ-1P11
X1 X1 D X1 turbo	V9.1DG V9.1-3DG V10.1DG	CZ-8KR CZ-8KR －－－－	CZ-8GR CZ-8GR －－－－	CZ-800P CZ-8PD2 CZ-80PK CZ-8PK2 ESC/P09	CZ-8PN1 MZ-1P10A MZ-1P11A MZ-1P17 ESC/P24

※DISKのMZ-80BF,MZ-1F07 やCZ-501F,CZ-801F,CZ-300F が必要
※DISKは、２ドライブ必要でMZ-80Bのみ70トラック仕様

文字種類	ＪＩＳ第一水準漢字2965文字＋非漢字　（ MZ-1R13,CZ-8KR 必要 ） ＪＩＳ第二水準漢字3384文字　（ ディスク内に内蔵 ） ユーザー外字　16×16＝80文字　24×24＝80文字 システム外字　16×16＝16文字　24×24＝227 文字
辞書機能	辞書３万語内蔵　＋　ユーザー辞書１万追加可能 自動学習機能付き（１万６千文字）－－－－－（MZ-80Bは、８千文字） 文書作成中にユーザー辞書登録・閲覧可能
変換機能	ローマ字変換／部首検索（ＪＩＳ第一・第二水準）／単語変換 ＪＩＳコード変換／文節変換／一括変換／半角変換
編集機能	挿入／削除／訂正／短文消去・登録・再生／枠あけ／枠とじ 倍角指定・解除／アンダーライン／罫線引き・消去 文書移動・複写・交換／左寄せ／中央寄せ／右寄せ／袋閉設定・解除 文書略図／禁則処理／改行設定／桁揃え／電卓計算／画面計算 上つき・下つき／タブ移動／作表機能／差し込み処理／辞書登録 語句登録／文頭移動／文末移動／機能一覧表示
文書書式	プリンタ指定／上部余白／下部余白／左側余白／右側余白 文字縦倍・普通・行単位／横書き・縦書き・行単位／文字間隔／行間隔
印刷機能	枚数指定／頁番号付け／部分印刷／差し込み印刷／連続再生印刷 再生間隔指定／自動供給機能／横書き／縦書き／倍角文字／半角文字 24ドットプリンタ完全対応　［ MZ-1P10 （ MZ-1P17 ）， MZ-1P11 ］
印刷能力	MZ-80BP5,CZ-800P －－－－－－－－－－－－ 最大横５０文字×８０行 MZ-80P6,MZ-1P07,ESC/P09,MZ-1P10,CZ8PN1 最大横６０文字×８０行 MZ-1P11,ESC/P24 －－－－－－－－－－－－－ 最大横９９文字×４０行
補助機能	文書初期化／データ複写／システム複写／旧文書再生／文書合成 文書挿入／文書削除／書式閲覧／機密解除／文書名再生／文書名変更 自動作表／外字作成（24/16）／外字一覧（24/16）／辞書閲覧／辞書登録 辞書削除／学習登録／学習消去　X1（ POP表示／ビデオ編集）
容　量	１文書　：　4000文字以内　（　１文書ファイルに35から 240頁　） １語句　：　126文字以内　（　１語句ファイルに1000語句　）

24ドットプリンタは、第二水準ROM ・カットシートフィーダコントロール可能

オールマシン語による超高速漢字住所録

検索速度　：　1000名中、１人検索時間最高50秒以内
並び替え　：　1000名並び替え時間　５分前後　150名並び替え時間　10秒前後
項　目　：　氏名、フリガナ、電話番号、住所１、住所２、住所３
郵便番号、備考１、備考２、備考３、備考４
熟　語　：　1200語（県名、市名 700語登録済）学習機能付　外　字　：　95文字
宛名印刷　：　シングルシール、ダブルシール、ハガキ印刷
検　索　：　２重条件検索方法
検索条件　：　無条件、同じ、大きい、小さい、含む、含まない
住所一覧　：　表示、印字は項目を自由に設定することが出来ます。（ディスク版）

機　種	TYPE	人数	熟語	部首	第二水準	品　番	価格	プリンタ
MZ-1500	ＱＤ	100	なし	あり	不可能	V8.3Q	15,000	MZ-1P08 MZ-1P14
MZ-80B MZ-2000,2200	DISK DISK TAPE ＱＤ	1000 1000 150 150	あり あり あり あり	なし なし なし なし	不可能 不可能 不可能 不可能	V5.3D V6.3D V6.3T V6.3Q	33,000 33,000 15,000 15,000	GP-500Z MZ-80P6 MZ-1P07
MZ-2000,2200	DISK	1000	あり	あり	内　蔵	V6.3DK	43,000	MZ-1P10
X1 D X1 X1 X1 turbo X1 turbo	DISK DISK TAPE DISK TAPE	1000 1000 150 1000 150	あり あり あり あり あり	なし なし なし なし なし	不可能 不可能 不可能 不可能 不可能	V9.3-3D V9.3D V9.3T V10.3D V10.3T	33,000 33,000 15,000 33,000 15,000	CZ-800P CZ-8PD2 CZ-80PK CZ-8PK2 EPSON

★　新発売商品　　　漢字ROM（ MZ-1R13 / CZ-8KR / MZ-1R23 ）が必要です。

商　品　名	機　種　名	TYPE	価　格
漢字ＢＡＳＩＣコンバータ	MZ-80B MZ-2000,2200 MZ-2000,2200	DISK DISK TAPE	￥18,000 ￥18,000 ￥13,000
漢字ファイル　（ KF-1 ）	MZ-80B MZ-2000,2200	DISK DISK	￥68,000 ￥68,000
ワープロ　ミニミニ ★ワープロ　ミニミニK （ MZ-1P17 専用 ）	MZ-2000,2200 MZ-2000,2200 MZ-1500 X1	TAPE ＱＤ ＱＤ TAPE	￥ 7,000 ￥ 8,000 ￥ 8,000 ￥ 7,000

※　ワープロ　ミニミニKは、横書・縦書・横倍・縦倍印刷ができます。
※　MZ-1P17 は、MZ-1P10 と上位コンパチです。

エレクトロ　ハウス
株式会社　スガヤ
〒416 静岡県富士市長通104-3　電話番号 (0545) 61-1417 （代）

お求めは全国マイコンショップ又は当店宛に現金書留（送料サービス）に機種名及びプログラム名を書いてお送り下さい。
営業時間　AM9:00～PM7:00
振込口座　清水銀行富士支店（当座）5683　定休日　毎週水曜日・第３日曜日

上記内容は、A3用紙にNEW miniでMZ-1P11 を使って印刷しました。

パーソナルコンピュータ

MZ-1500 標準価格 89,800円

●キャラクタづくりも自在、強力グラフィック機能●音楽演奏もお手のもの、充実のサウンド機能●上達に合わせて進化するクリーン設計●能力をグンとアップさせるRAMファイル(オプション)●おしゃべりもOK、ボイスボード(オプション)●買ったその日から即使える実用ソフトつき。

▲写真の14型カラーディスプレイCU-14F1B標準価格64,800円はオプションです。●CRT画面はハメコミ合成で「ギャラガ」(©㈱ナムコ)より。またその他の画面は「ロードランナー」(ユニバース)、「ドアドアmkII」(エニックス)、「ユーカラJJ」(東海クリエイト)、「ミュージックダンス」(ロータス)、「サンダーフォース」(テクノソフト)より。※ロードランナーはUSA Broderbund Software Incの登録商標です。

●新作ソフトも続々登場、いよいよ充実してきたQDアプリケーション

ソフトジャンル	ソフト名称	標準価格(円)	ソフトメーカー
ゲーム	チャンピオンシップロードランナー※	5,200	ユニバース(コスモス岡山)
	MILKY WAY	4,800	マイクロネット
	ウォーリィ	4,800	マイクロキャビン
	ギャラガ	4,500	電波新聞社
	マジックファクトリー	5,800	コムパック
	信長の野望	5,800	光栄
	野球狂	5,800	ハドソン販売

ソフトジャンル	ソフト名称	標準価格(円)	ソフトメーカー
ゲーム	サンダーフォース	4,800	テクノソフト
	フラッピー	4,800	デービーソフト
ビジネス	ユーカラJJ(MZ-1P17専用)	12,800	東海クリエイト
	NEW VIP	12,800	デービーソフト
学習	中学数学シリーズ	各 4,500	数研塾
	中学英作文シリーズ	各 7,800	
	中学・高校社会科シリーズ	各 4,500	

●上記のソフトはほんの一例です。詳しくは"MZ APPLICATION NEWS"をご覧ください。※ロードランナーはUSA Broderbund Software Incの登録商標です。

シャープ株式会社 本社 〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)●お問い合わせは…本社内国内情報システム営業本部まで。

資料請求券 MZ-1500 Oh!MZ 1枚

Oh!MZ ㈱日本ソフトバンク発行 Printed in Japan 定価480円 雑誌02179−1

昭和60年5月14日国鉄首都特別扱承認雑誌第8269号 昭和58年11月2日第三種郵便物認可